VYSOKÁ ŠKOLA BÁŇSKÁ – TECHNICKÁ UNIVERZITA OSTRAVA

UNIVERZITNÍ STUDIJNÍ PROGRAMY

Řízení mobilního robotu

Control of the Mobile Robot

BAKALÁŘSKÁ PRÁCE

Student: Andrey Vasilev

Vedoucí bakalářské práce: prof. Dr. Ing. Petr Novák

Ostrava 2011

VŠB - Technická univerzita Ostrava
Univerzitní studijní programy
Katedra robototechniky

Zadání bakalářské práce

Student: **Andrey Vasilev**

Studijní program: B3943 Mechatronika

Studijní obor: 3906R006 Mechatronické systémy

Téma: Řízení mobilního robotu
Control of the Mobile Robot

Zásady pro vypracování:

1. Seznamte se z vlastnostmi robotu UMU28A s mikrokontrolérem PICAXE-28X1.
2. Seznamte se z vývojovým prostředím PICAXE.
3. Navrhněte soubor úloh typu sledování čáry.
4. Navrhněte úlohu typu "minisumo" pro dvojici robotů.
5. Vytvořte vhodný manuál.

Poznámka: práci též doložte v elektronické podobě ve formátu MS Word.

Seznam doporučené odborné literatury:

1. PICAXE manual2 - Basic Commands, Version 6.7 01/2009. Součást vývojového prostředí programu.
2. PIC16F886 DataSheet, Microchip. [online] Dostupné z <http://www.microchip.com>
3. MiniSumo - zápas robotů. [online] Dostupné z <http://www.hobbyrobot.cz/mini_sumo.htm>
4. NOVÁK, P. *Mobilní roboty – pohony, senzory, řízení.* 1.vydání, Praha: BEN – technická literatura, 2005, 248 stran. ISBN80-7300-141-1.

Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí bakalářské práce: **prof. Dr.Ing. Petr Novák**

Datum zadání: 17.12.2010

Datum odevzdání: 23.05.2011

prof. Dr.Ing. Petr Novák
vedoucí katedry

prof. Ing. Petr Noskievič, CSc.
prorektor pro studium

Místopřísežné prohlášení autora

Prohlašuji, že jsem celou bakalářskou práci *Řízení mobilního robotu* včetně příloh vypracoval samostatně pod vedením prof. Dr. Ing. Petra Nováka a uvedl jsem všechny použité literární a odborné zdroje.

V Ostravě 19. 5. 2011

.....

vlastnoruční podpis autora

Prohlašuji, že

- jsem byl seznámen s tím, že na moji bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb., autorský zákon, zejména § 35 – užití díla v rámci občanských a náboženských obřadů, v rámci školních představení a užití díla školního a § 60 – školní dílo.
- beru na vědomí, že Vysoká škola báňská – Technická univerzita Ostrava (dále jen „VŠB-TUO“) má právo nevýdělečně ke své vnitřní potřebě bakalářskou práci užít (§ 35 odst. 3).
- souhlasím s tím, že bakalářská práce bude v elektronické podobě uložena v Ústřední knihovně VŠB-TUO k nahlédnutí a jeden výtisk bude uložen u vedoucího bakalářské práce. Souhlasím s tím, že údaje o kvalifikační práci budou zveřejněny v informačním systému VŠB-TUO.
- bylo sjednáno, že s VŠB-TUO, v případě zájmu z její strany, uzavřu licenční smlouvu s oprávněním užít dílo v rozsahu § 12 odst. 4 autorského zákona.
- bylo sjednáno, že užít své dílo – bakalářskou práci nebo poskytnout licenci k jejímu využití mohu jen se souhlasem VŠB-TUO, která je oprávněna v takovém případě ode mne požadovat přiměřený příspěvek na úhradu nákladů, které byly VŠB-TUO na vytvoření díla vynaloženy (až do jejich skutečné výše).
- beru na vědomí, že odevzdáním své práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb., o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších předpisů, bez ohledu na výsledek její obhajoby.

V Ostravě 19. 5. 2011

.....
vlastnoruční podpis autora

Jméno a příjmení autora práce: Andrey Vasilev

Adresa trvalého pobytu autora práce: Ostrava - Bělský Les, B. Václavka 7

Poděkování

Rád bych poděkoval prof. Dr. Ing. Petru Novákovi za cenné rady a připomínky, také za poskytnuté materiály a školení, čímž velice pomohl vypracování této bakalářské práce.

Abstrakt

Předmětem bakalářské práce *Řízení mobilního robotu* je seznámení se s mobilním robotem *UMU28A* a následné použití získaných vědomosti k návrhu manuálu a souboru úloh pro zmíněný školní mobilní robot. První část se zabývá seznámením se s mobilním robotem *UMU28A*, konkrétně senzory, akčními členy a mikrokontroléry řídicími mobilní robot. Druhá část se zabývá seznámením se s vývojovým prostředím *PICAXE Programming Editor* určenému k návrhu programů zadaných řešených úloh pro mobilní robot *UMU28A*. Třetí část se zabývá popisem způsobu využití senzorů a akčních členů, také popisem struktur programů řešících zadané úlohy.

Abstract

The subject of Bachelor Degree Thesis 'Control of the Mobile Robot' is to master mobile robot *UMU28A* and consequential usage of obtained knowledge to design manual and task set for mentioned mobile robot. The first part is dealing with mastering of mobile robot *UMU28A*, to be specific sensors and actuating units and microcontrollers in control of mobile robot. The second part is dealing with mastering of development tool *PICAXE Programming Editor* which is designed for development of demanded task set for mobile robot *UMU28A*. The third part is dealing with description of concept for usage of sensors and actuating units, also description of programs structural models for solving demanded task set.

Klíčová slova

Mobilní robot, programování mikrokontroléru, mikroprocesor PICAXE-28X1, úlohy pro UMU28A

Key words

Mobile robot, programming of microcontroller, microprocessor PICAXE-28X1, tasks for UMU28A

Obsah

Seznam použitých zkratek, označení a termínů

AAA	– velikost článku/ akumulátoru (mikrotužka)
ADC	– analogově digitální převodník (Analog-Digital Converter)
Basic	– programovací jazyk
Black Box	– zařízení, systém nebo objekt, jehož vnitřní fungování neznáme
CMOS	– technologie výroby integrovaných obvodů
GM8PW	– malý stejnosměrný motor s převodovkou
IR	– infračervené (Infra-Red)
L293D	– čtyřnásobný půlmost H
LED	– fotodioda (Light Emitting Diode)
PC	– osobní počítač (Personal Computer)
PIC 16F688	– mikrokontrolér od firmy Microchip
PICAXE-28X1	– mikrokontrolér od firmy Microchip
PWM	– pulzní šířková modulace (Pulse Width Modulation)
QRD1114	– senzor intenzity infračerveného záření s analogovým výstupem
SFH5110	– senzor infračerveného záření s diskrétním výstupem
TTL	– standart používaný u integrovaných obvodů
UMU-01	– podvozek pro malé mobilní roboty
USBCOM	– převodník USB/RS232
breakpoint	– bod pozastavení simulace
line-by-line	– řádek po řádku

Úvod

Mobilním robotem rozumíme mechatronický systém s určitou mírou samostatnosti, který je schopný se přemisťovat. Robot dle účelu předepsaným způsobem vykonává zadané úkoly při interakci s okolním prostředím dle zadaných priorit. Cílem programátora mobilního robotu je program schopný řešit interakci mobilního robotu s okolím, čím efektivnější reakce robotu budou zapotřebí, tím větší budou nároky na výpočetní výkon řídicího členu robotu, v daném případě mikrokontroléru *PICAXE-28X1*.

Robot vnímá okolí pomocí senzorů a reaguje pomocí akčních členů, které můžou, ale nemusí být řízeny přímo řídicím členem robotu. Rychlost rozhodování ovlivňuje doba potřebná k měření a zpracování informace, rychlost reakce pak je ovlivněna komunikačními prodlevami mezi řídicím členem a členem řídicím akční člen. Toto platí v případech, kdy akční člen není přímo připojen k členu řídicímu.

Pro zabezpečení efektivního vykonávání úkolů požadovaných po mobilním robotu je zapotřebí co nejlepší informovanost o prostředí, ve kterém se robot bude pohybovat. V případě nedostatku informaci o prostředí musí programátor pracovat s předpoklady o informacích poskytovaných senzory mobilního robotu. V dané prácí se pracuje s požadavkem plné autonomnosti mobilního robotu založené při znalosti prostředí v interakci.

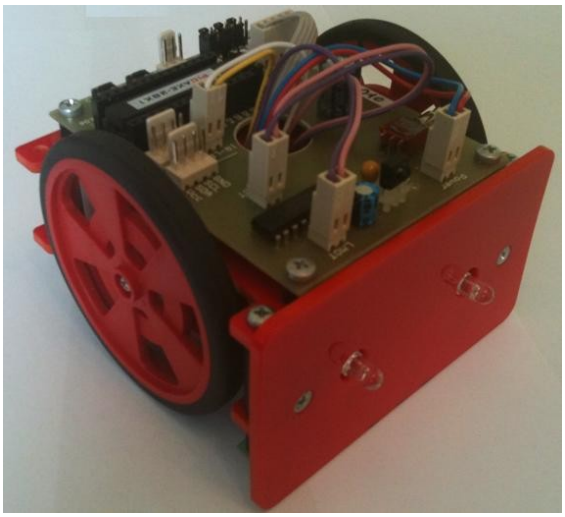
Programování mobilního robotu s kontrolérem *PICAXE-28X1* je vhodné i pro začátečníky, jelikož je použit zaváděcí program v kontroléru umožňující nahrání programu do kontroléru pomocí sériové linky bez použití překladače. Programování je možné pomocí jazyka Basic.

1 Robot UMU28A a mikrokontrolér PICAXE-28X1

1.1 Robot UMU28A

Jedná se o mobilní robot určený pro jednoduché aplikace typu *sledování čáry, minisumo* a další. Robot je vyráběn a dodáván firmou *Snail Instruments* [1]. *Obr. 1*, *Obr. 7*, *Obr. 8*, *Obr. 9*, *Obr. 10*

Robot Umík využívá podvozku *UMU-01*. Jako senzorů pro detekci okraje arény v aplikaci *minisumo* nebo pro sledování čary je využito senzorů *QRD1114*. Pro detekci překážky případně soupeře je využit senzor *SFH5110* s využitím dvou *IR – LED20* diod. Jako řídicí elektronika slouží mikroprocesor *PICAXE-28X1* jenž je vybaven speciálním zaváděcím programem dovolujícím jeho programování v *Basicu*. Robot obsahuje samostatný procesor řízení motorů pomocí *PWM* , řízení samostatného procesoru pomocí sériového kanálu. Budič motorů využívající můstek *L293D*. K napájení slouží 6 akumulátorů velikosti *AAA*. Pro zavádění programu není třeba překladač, pro zavádění programu se využívá sériový port nebo *USBCOM3* [1].



Obr. 1 − mobilní robot UMU28A

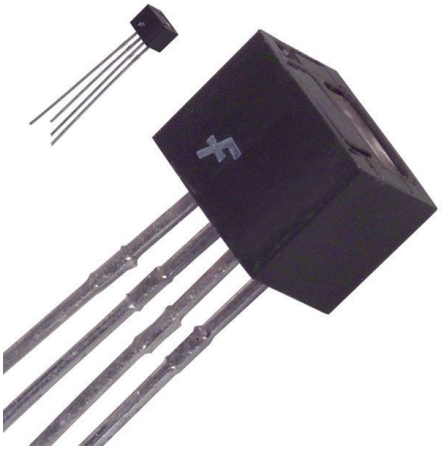
1.1.1 Senzor IR záření SFH5110

Jedná se o infračervený přijímač vhodný k detekci soupeře či překážek, komunikaci s *PC*, ovládání robotu. Senzor je vysoce citlivý a je kompatibilní s technologiemi *TTL* a *CMOS*. Vyžaduje zdroj *IR* signálu modulovaného 38kHz (*IR-LED* nebo dálkové ovládání). Pouzdro je černé a slouží k filtraci denního světla. Senzor je odolný na rušení signály o jiných frekvencích (pásmová propusť) [2]. *Obr. 2*

Obr. 2 – infrasenzor SFH5110, obrázek převzat z [2]

1.1.2 Senzor IR záření QRD1114

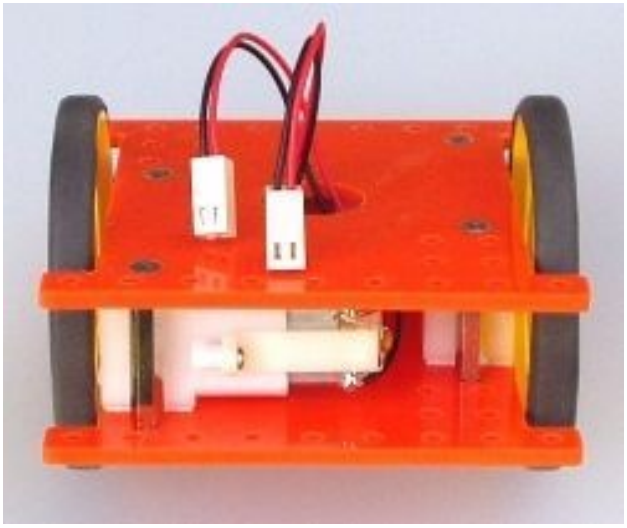
Jedná se o odrazový infrasenzor vhodný pro aplikaci sledování čáry či detekci okraje arény. Senzor je složen z IR−LED a fototranzistoru, pouzdro má 4 vývody. Maximální vzdálenost odrazového materiálu činí 6mm a při dynamické změně této vzdálenosti se výsledky měření tohoto senzoru nelineárně mění, závislost změny je uvedena v [3]. *Obr. 3*



Obr. 3 – odrazový infrasenzor QRD1114,
obrázek převzat z [3]

1.1.3 Podvozek mobilního robotu UMU-01 a akční členy

UMU-01 je univerzální pohonná jednotka obsahující 2 stejnosměrné motorky s převodovkou (*GM8PW*), které jsou vybaveny odrušovacími kondenzátory, kablíky a konektory. Součástí podvozku je též ocasní opěrná všesměrová kulička a radlice [4]. *Obr. 4*



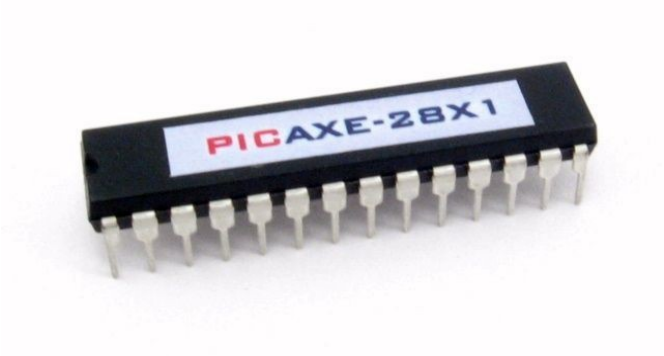
Obr. 4 – podvozek UMU-01, obrázek převzat z [4]

1.2 Mikrokontrolér PICAXE-28X1

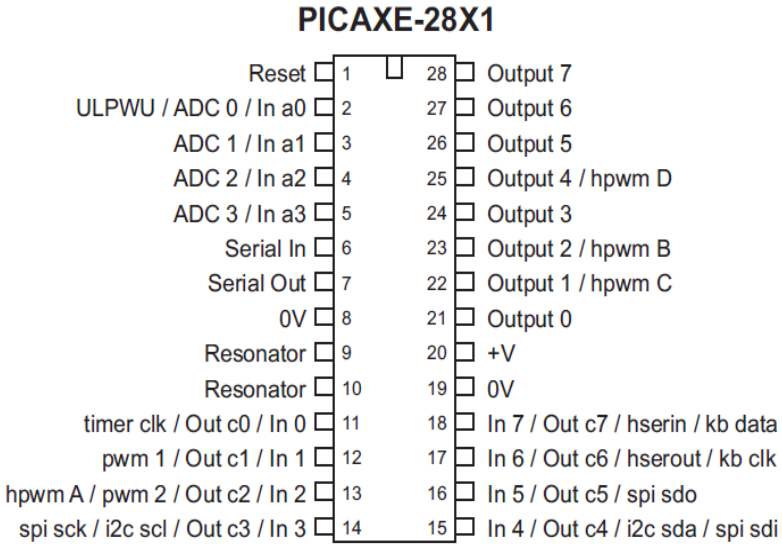
PICAXE-28X1 je kontrolér z rodiny PIC, vyráběné firmou *Microchip*, naprogramovaný speciálním zaváděcím programem dovolujícím programování kontroléru pomocí nenákladného programu, tzn., že celý systém pro nahrání programu sestává pouze z obyčejného sériového kabelu, standard RS232 nebo USB. Proces přeprogramování kontroléru je rychlý díky jeho předprogramování. ***Obr. 5***

K programování kontrolérů PICAXE se používá programovací jazyk *Basic*, případně grafické programování pomocí vývojových diagramů, které jsou použitelné jen pro některé kontroléry řady PICAXE. Ladění hotových programů je jednodušší, nežli v programovacích jazycích *assembleru* nebo *C*.

Kontrolér má hardwarovou podporu komunikačního protokolu I2C, pro snadné periferní připojování, také obsahuje vnitřní časovač a zvýšený počet proměnných a rozdělitelnou zápisníkovou paměť, vylepšené matematické funkce aj. [5].



Obr. 5 − PICAXE-28X1, obrázek převzat z [4]



Obr. 6 − rozložení pinů mikrokontroléru PICAXE-28X1, obrázek převzat z [5]

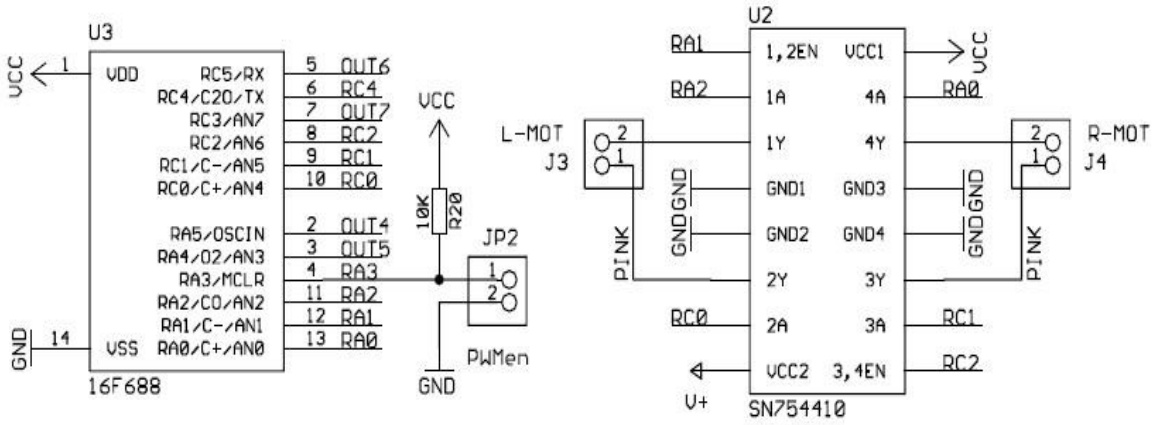
1.2.1 Vlastnosti kontroléru PICAXE-28X1

Kontrolér *PICAXE-28X1* je založen na technologii kontroléru *PIC 16F886 IC*. Počet řádku paměti programu činí 1000 řádku. Počet vstupů činí 0÷12, počet výstupu 9÷17. Počet *ADC* činí 0÷4 (8/10bitové). Obvyklá operační rychlost činí 4MHz, maximální operační rychlost může dosahovat 20MHz. Kontrolér podporuje *line-by-line* simulaci a přerušení. ***Obr. 6***

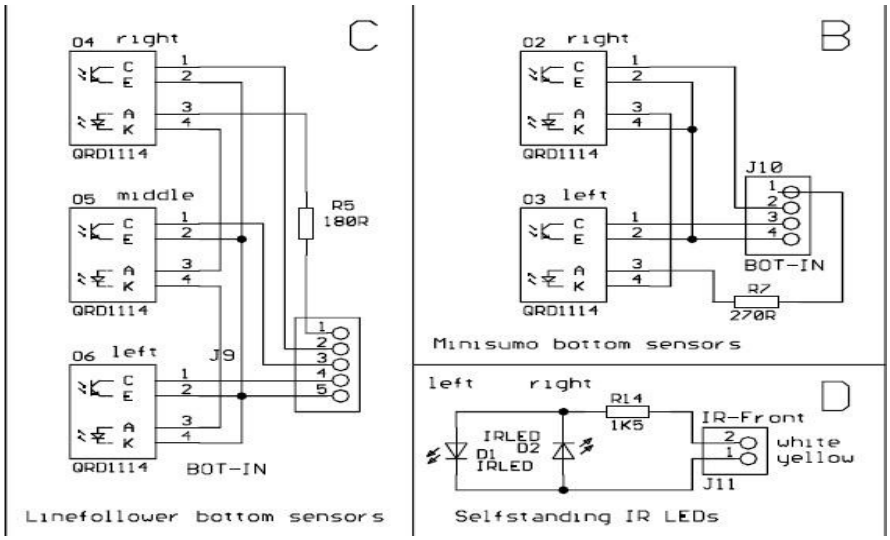
Kontrolér ovládá funkce dovolující jako vstupní zařízení klávesnici, vysílaní a příjem pomocí *IR* záření, přehrávaní uživatelem definovaných zvukových tónů, řízení motorů pomocí PWM, čítání vstupních pulsů, sériovou komunikaci i s PC.

Mobilní robot také dovoluje připojení dalších externích akčních členů nebo senzorů v případě potřeby, k tomuto účelu slouží sběrnice I2C (až 16 dalších zařízení).

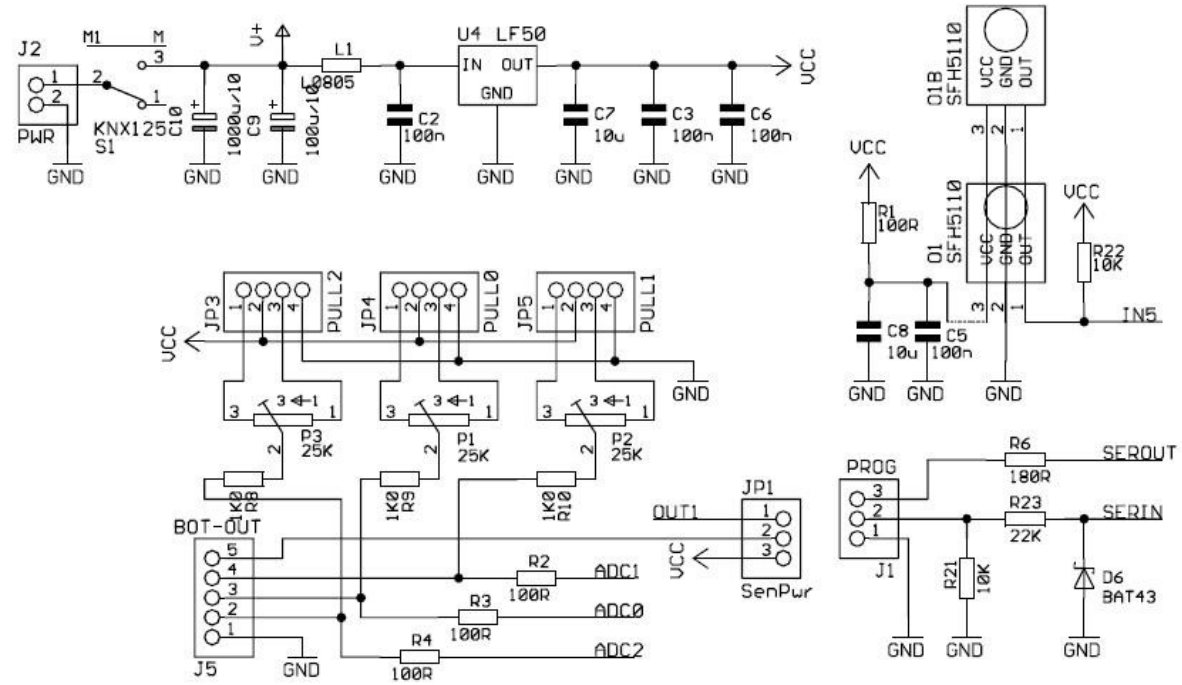
1.2.2 Schémata zapojení



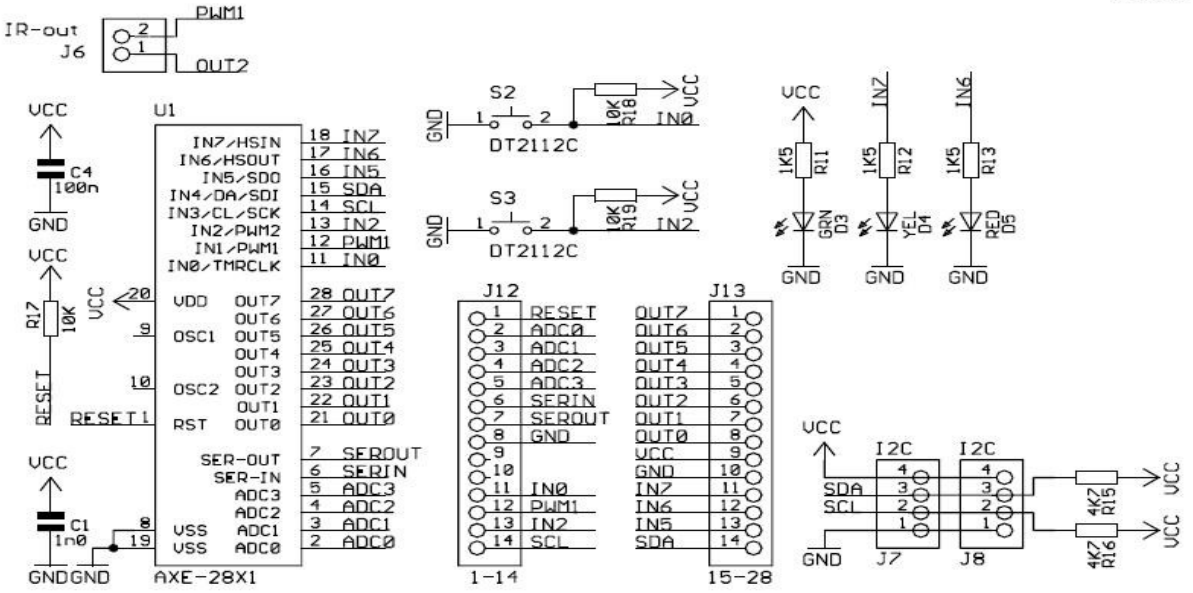
Obr. 7 – schéma zapojení motorků k H-mostu a mikrokontroléru PIC 16F688 jej ovládajícího, obrázek převzat z [7]



Obr. 8 – schéma zapojení infrasenzorů QRD1114 (C) a IR-LED diod (B), obrázek převzat z [7]



Obr. 9 – schéma zapojení tří trimrů (regulují citlivost spodních senzorů), senzorů IR záření SFH5110 a sériové linky (a napájení robotu), obrázek převzat z [7]



Obr. 10 – schéma zapojení mikrokontroléru PICAXE-28X1 , spínačů, informačních LED diod, napájení IR- LED a I2C sběrnice, obrázek převzat z [7]

2 Vývojové prostředí PICAXE

Vývojovým prostředím se rozumí program, ve kterém bude program pro ten či onen program pro různé aplikace vyvíjet, neboli psát a testovat program. Program *PICAXE Programming Editor* je volně dostupný na stránkách výrobce [5], [2], je také dodáván k zakoupeným výrobkům firmy *Snail Instrument* obsahujícím kontrolér řady PICAXE.

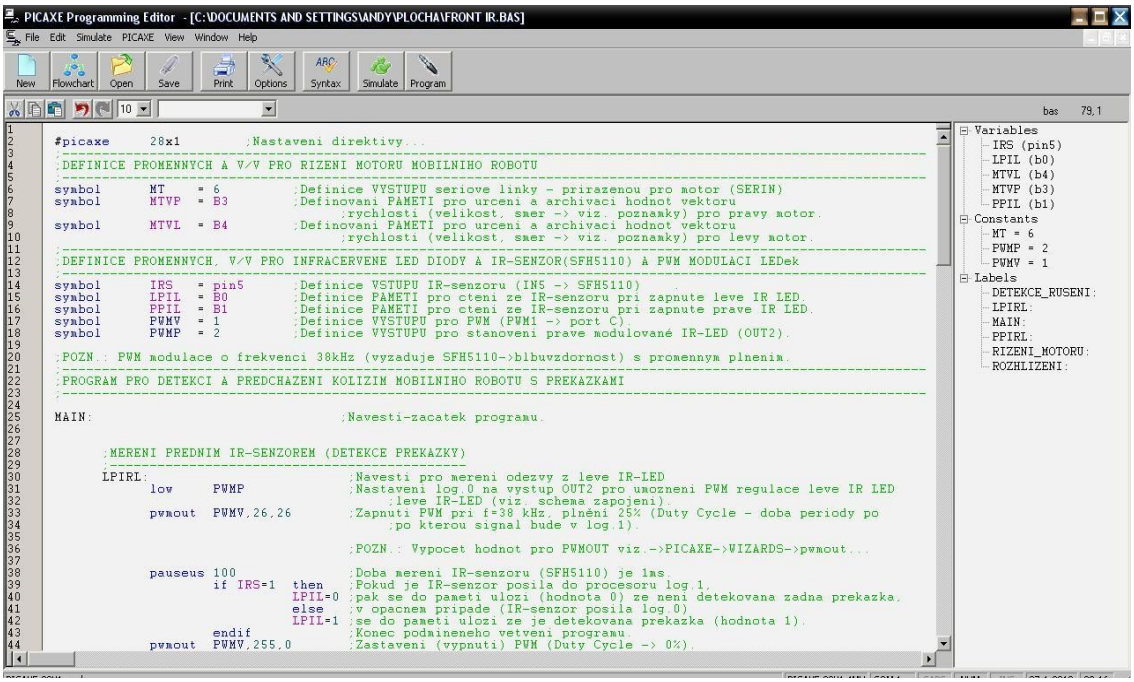
Vývojové prostředí obsahuje vše potřebné ke zprovoznění programu pro mikrokontroléry řady PICAXE. V daném případě se jedná o verzi 5.2.2. ***Obr. 11***

2.1 Programování pomocí jazyka Basic

„BASIC je rodina programovacích jazyků vysoké úrovně, která byla zavedena jako jednoduchý nástroj pro výuku programování. K jednoduchosti přispívalo i to, že klíčová slova jazyka vychází z běžné angličtiny. Jazyk navrhli v roce 1963 John G. Kemeny a Thomas E. Kurtz z Dartmouthské univerzity (Hanover, New Hampshire). Název BASIC je zkratkou anglických slov Beginner's All-purpose Symbolic Instruction Code.“ [8]

Vývojové prostředí obsahuje celkový seznam použitelných příkazu programovacího jazyka Basic v záložce Help, příkazy jsou také k nalezení na stránkách výrobce [5]. Oproti programování v *Assembleru* je programování v *Basicu* jednodušší, což je vykoupeno omezením možnosti ovládání mikrokontroléru.

Již zmíněnou výhodou je zaváděcí program obsažený v mikrokontrolérech PICAXE dovolující nahrávaní programu do kontroléru bez překladače. Konverze programu na strojový kód je ošetřena zaváděcím programem kontroléru řady PICAXE.

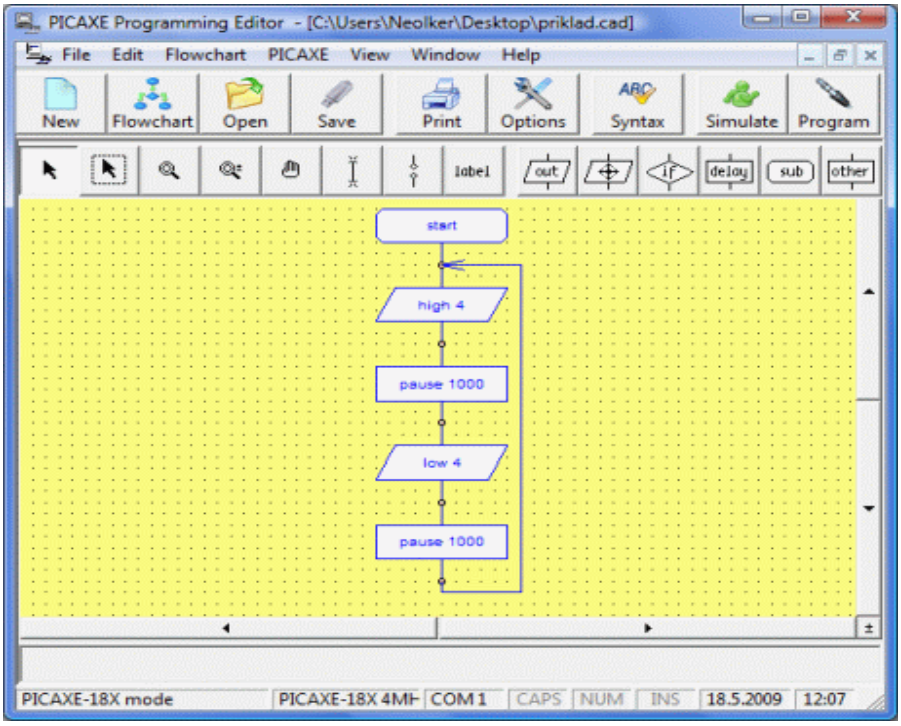


Obr. 11 – okno programovacího prostředí PICAXE Programming Editor, programování pomocí programovacího jazyku Basic

2.2 Programování pomocí Blokových schémat

„Schéma bloková - funkčný diagram, štruktúrny obraz - toplogicko-geometrické znázornenie štruktúry systémov. Každý prvok blokovej schémy je spravidla čiernou skrinkou a zvykne sa zobrazovať ako jednoduchý geometrický obrazec (stvorec, obldĺžnik, trojuholník, kruh atď.). Spôsob spojenia takychto prvkov do jedného komplexného systému býva znázorňovaný šipkami, ktoré udávajú smer pôsobenia príslušných prvkov na seba, smer informačného toku od jedného prvku systému k druhému.“ [9]

Programování pomocí blokových schémat je přehlednější než programování pomocí *Basicu* avšak nadále redukuje možnosti ovládání kontroléru pomocí programu. Nespornou výhodou je však možnost tvořit a opakovaně užívat předdefinované funkční bloky. *PICAXE-28X1* programování pomocí blokových schémat nepodporuje. *Obr. 12*



Obr. 12 – okno programovacího prostředí PICAXE Programming Editor, programování pomocí blokového schématu (Flowchartu)

2.3 Nástroje ladění programů

Po napsání programu je nutné jej zkontrolovat na chyby (zápis/přiřazení proměnných nebo špatně napsané/definované příkazy) pomocí funkce SYNTAX. Kontrola na chyby je nutná před každým nahráním programu do kontroléru, budou-li nalezeny chyby v programu, objeví se okno na ně odkazující.

Nutnost odstraňování chyb a případné úpravy neboli ladění programu vyplývá z požadavku na bezpečnost programovaného mobilního robotu, jelikož přehlednutá chyba může zapříčinit jeho poškození a tím i finanční ztráty za opravy nebo případné pořízení nového mobilního robotu. Vývojové prostředí PICAXE Programming Editor poskytuje dva nástroje napomáhající eliminaci chyb a ladění programu.

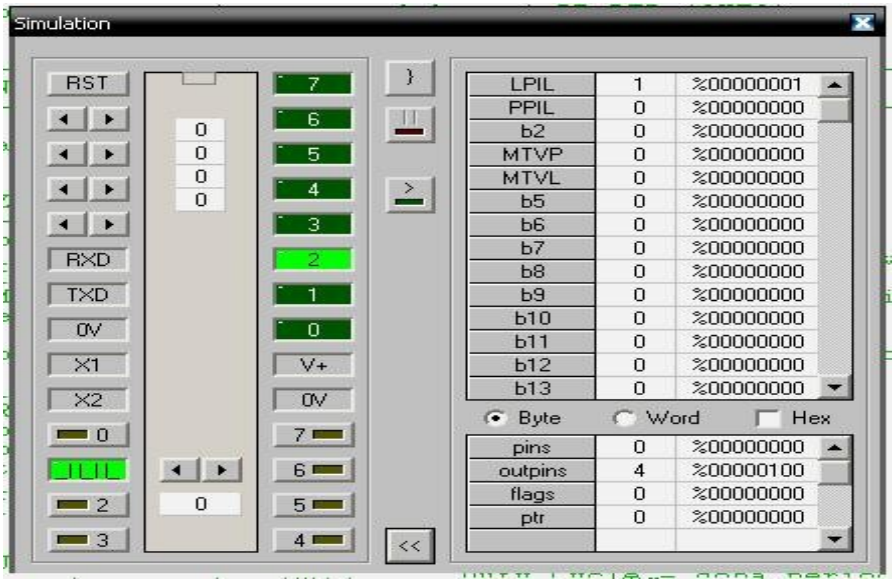
2.3.1 Simulace

„Simulace se používá v mnoha souvislostech, zahrnujících modelování přírodních systémů nebo lidských systémů s cílem získat poznatky o jejich fungování. Jiné souvislosti zahrnují technologické simulace pro optimalizaci výkonu, bezpečnostní inženýrství, testování, školení a vzdělávání. Simulace může být použita pro zobrazení případných reálných dopadů alternativních podmínek a způsobů jednání.

Klíčové otázky v simulaci zahrnují např. pořízení platných zdrojů informací o příslušném výběru klíčových charakteristik a chování, využití zjednodušujícího odhadu a předpokladů v rámci simulace a věrnost a platnost výsledků dané simulace.“ [10]

Po kontrole pomocí funkce SYNTAX je možné spustit simulaci programu pro zjištění možných nesouladů ve fungování programu je možné je opravit a následně spustit simulaci. Simulace je dostupná jak krokováním tak automatická s nastavitelnými *Breakpointy*. Okno zobrazené na *Obr. 13* dovoluje nastavovat obsah paměťových proměnných stavy (pro analogové vstupy → hodnoty 0-255) na vstupních i výstupních vývodech kontroléru, zapínat a vypínat *PWM* modulaci a také sledovat přenos dat sériovou linkou, případně provést restart simulace.

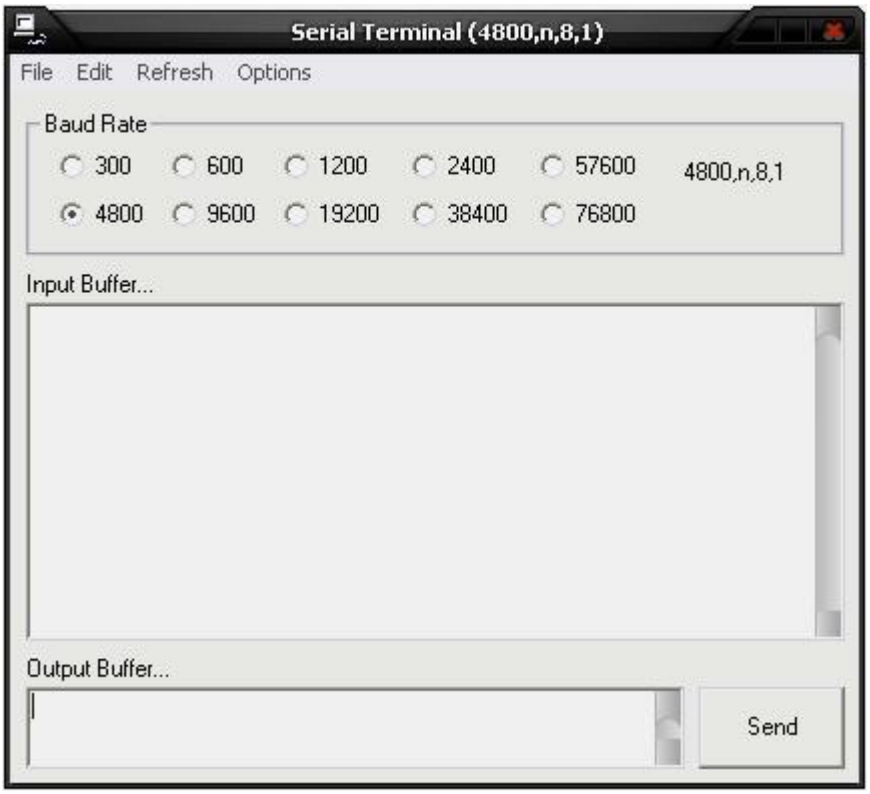
Jako prostředek k odstraňování chyb je simulace velmi výhodná jelikož dovoluje předcházet kritickým chybám, které by se mohly projevit i poškozením mobilního robotu. K provádění simulace není třeba mít připojen stroj nebo robot k PC.



Obr. 13 − okno (Variables) řídicí a monitorující paměť a stavy na vývodech procesoru při simulaci (PICAXE-28X1)

2.3.2 Sériová komunikace

K ladění programu je velice užitečná schopnost kontroléru PICAXE-28X1 komunikovat pomocí sériového rozhraní typu RS232. Toto umožňuje přímou komunikaci mezi vývojovým prostředím a kontrolérem přidáním několika příkazů do programu kontroléru, tzn. poskytnutí možnosti sledovat chování mobilního robotu v závislosti na jeho programu a tím detekovat nesoulad neboli chybu v programu nahraném do procesoru PICAXE-28X1, v závislosti na požadavcích, které má daný program splňovat (úloha, kterou má plnit). ***Obr. 14***



Obr. 14 – okno terminálu pro komunikaci mezi kontrolérem a PC

Vývojové prostředí má celou řádu dalších funkcí napomáhajících programování kontroléru, avšak pro jejich použití jsou potřeba znalosti z oblasti elektroniky, číslicové techniky a základní znalosti automatizace.

3 Využití senzorů a řešení souboru úloh

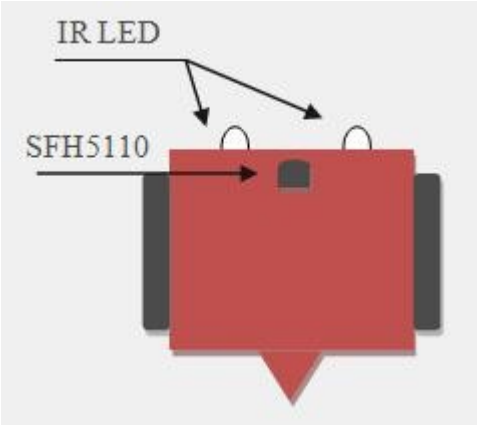
Před psaním programu je důležité obeznámit se o způsobu měření senzory daného mobilního robotu a jejich možnostmi. Důležité jsou dosahy, rozsahy a druh poskytnuté informace senzorů. Myšleno do jaké vzdálenosti dokáže detekovat překážku a jak například ovlivní typ povrchu, od něhož se *IR* záření bude odrážet, jeho odráženou intenzitu a disperzi a následně v jaké formě poskytuje senzor data kontroléru v diskrétní či analogové.

3.1 Měření senzory

Robot využívá senzoru jako organu vnímání a čím jsou přesnější, rychlejší a početnější tím bude informovanost robotu o okolí lepší a tím efektivněji se bude moct rozhodnut o dalších akcích v závislosti na prioritách zadaných v programem řešícím aplikaci. U mobilního robotu je dobré povědomí o možnostech senzorů ještě důležitější jelikož při kritické situaci musí se s problémy vypořádat sám, jelikož vyšší úroveň rozhodování neboli operátor může být z rozhodovacího řetězce vyloučen, pro plně autonomní roboty.

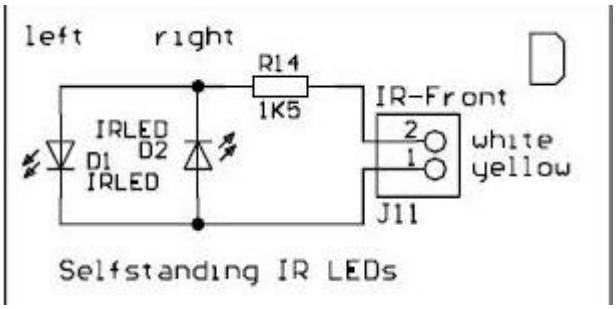
3.1.1 Měření senzorem SFH5110

Senzor infračerveného záření s diskrétním výstupem je umístěn na horní přední části mobilního robotu uprostřed *Obr. 15*. Senzor je odolný vůči rušení z jiných zdrojů *IR* záření, jelikož snímá záření vysílané v pulsech o frekvenci 38kHz, čímž se eliminuje vysílané záření v pulsech o jiné frekvenci.

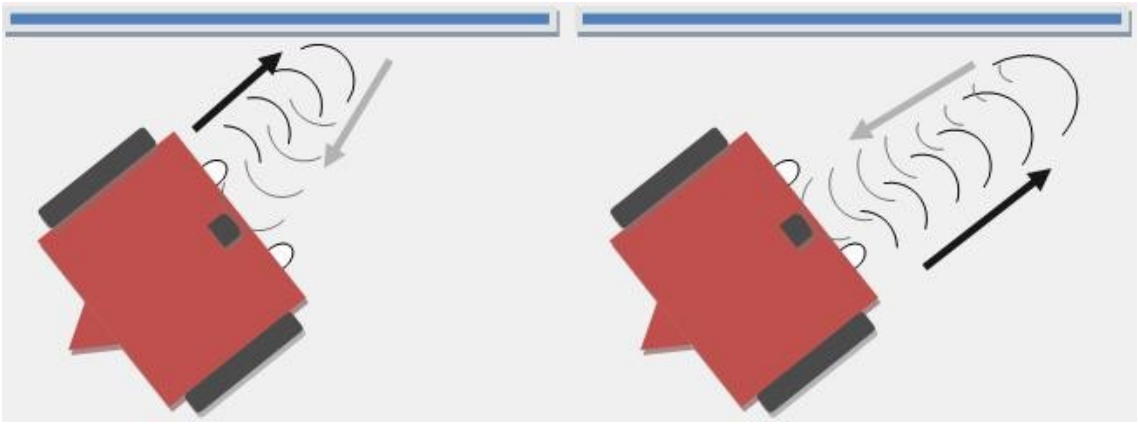


Obr. 15 – umístění senzoru SFH5110 a IR−LED diod (pohled shora)

Měření se provádí střídavě pro detekci odrazu z pravé a levé stran, neboli nejdřív se vysílá pomocí PWM modulace záření z levé diody zatím co pravá dioda je vypnutá a senzor snímá odražené záření a ukládá v paměti mikrokontroléru. Následně se vysílá pomocí *PWM* modulace záření z pravé diody zatím co levá dioda je vypnutá a senzor snímá odražené záření a ukládá v paměti mikrokontroléru. ***Obr. 17***, ***Obr. 16***



Obr. 17 – zapojení IR LED diod (pin č. 2 → PWM1 pin č. 12, pin č.1→OUT2 pin č. 13)

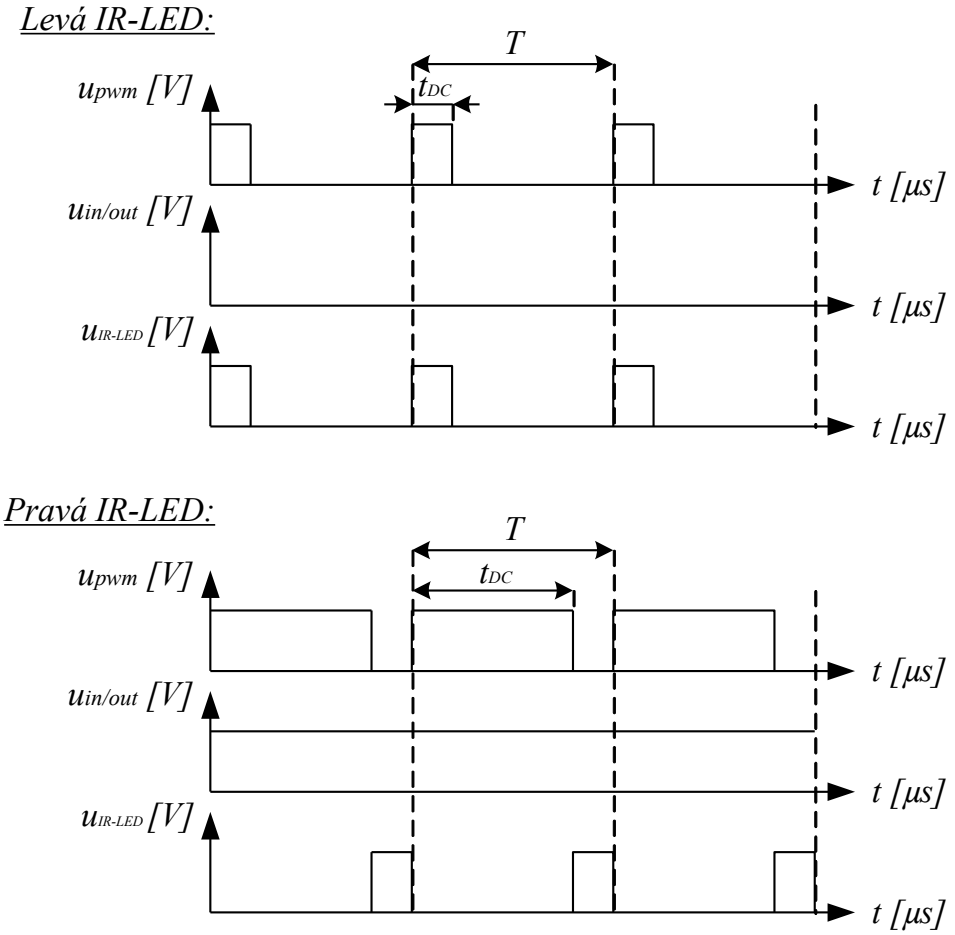


Obr. 16 – detekce polohy překážky vůči robotu UMU28A senzorem SFH5110 (pohled shora), detekce z levé strany (vlevo), detekce z pravé strany (vpravo)

Střídavé měření odrazu pro určení polohy překážky před robotem je podmíněna počtem senzorů, v tomto případě jedním, a zapojením *IR−LED* diod *Obr. 17*. Z obrázku je vidět, že diody jsou zapojeny antiparalelně, což znemožňuje buzení obou diod zároveň.

Zapojení pinu *číslo 2* (white → ***Obr. 17***) je provedeno na pin kontroléru *číslo 12* určený pro *PWM* modulací (***Obr. 6***, ***Obr. 10***). Zapojení pinu číslo 1 (yellow → ***Obr. 17***) je provedeno na pin kontroléru *číslo 13* (***Obr. 6***, ***Obr. 10***) schopný dle potřeby plnit funkcí jak vstupního, tak výstupního diskrétního pinu. Pro měření odrazu záření z levé (left) *IR-LED* je buzení pomocí PWM přivedeno na pin kontroléru č. 12 (2 white pin ***Obr. 17***) a pin č. 13 je nastaven jako vstupní. Plnění PWM pak určí výkon záření *IR-LED*, pro toto měření je plnění 25 %. Pro měření odrazu záření z pravé (left) *IR-LED* je buzení pomocí PWM přivedeno na pin kontroléru č. 12 (2 white pin ***Obr. 17***) a pin č. 13 je nastaven jako vstupní. Plnění PWM pak určí výkon záření *IR-LED*, pro toto měření je plnění 75 %. Průběhy buzení diod jsou znázorněny *Obr.* ***18***.

Doba periody (*T*) a její plnění (t_{DC}) lze nastavit, pro výpočet potřebných hodnot je možné použít funkci vývojového prostředí *PWMOUT Wizard*. K řízení a iniciaci PWM se použije příkaz *pwmout*, kde pomocí tří hodnot se nastaví doba trvání periody a činitel plnění periody. Maximální trvání periody je 256μs pro vnitřní frekvenci kontroléru 4MHz, pro každé další navýšení vnitřní frekvence kontroléru se maximální trvání periody úměrně zmenší (např.: 8MHz → 123μs).



Obr. 18 – Průběhy napětí na pinech kontroléru a příslušné buzení IR-LED diodě (upwm → pin kontroléru č. 12, uin/out → pin kontroléru č. 13, uir-led → příslušná IR-LED dioda)

Informace k následnému vyhodnocení polohy překážky před mobilním robotem jsou diskrétní, neboli dvoustavové, což je poměrně omezená informace udávající jen zda je nebo není překážka detekována, informace neudává vzdálenost mobilního robotu od detekované překážky. V úvahu se musí brát doba trvání měření, která je relativně velká ve srovnání s dobou následného vyhodnocení informace procesorem.

Řešit problém určení přibližné vzdálenosti od překážky je možné a řešení je závislé na intenzitě odraženého záření. Zatím je možné určit jen dvě polohy vzdálenosti mobilního robotu vůči překážce a to před a za hranicí dosahu detekčního pásma senzoru SFH5110. Intenzita odráženého záření je závislá na třech hlavních faktorech.

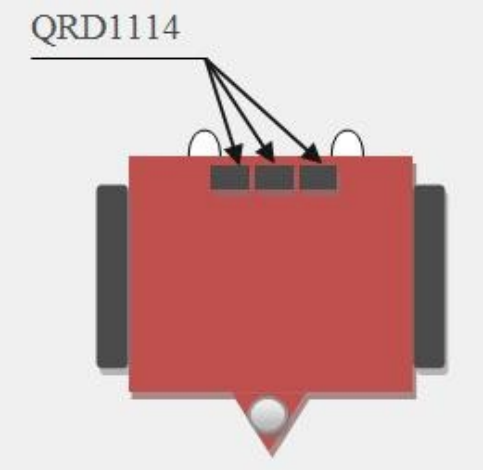
Prvním faktorem je intenzita záření *IR−LED* diod respektive jejich vyzářený výkon, které je nastavitelné díky použití *PWM* modulace, která dovoluje regulovat intenzitu záření *IR−LED* diod neboli jejich výkon.

Druhým faktorem je tvar, barva a lesklost povrchu. Jelikož matné a tmavší povrchy pohlcují více záření, stejně jako povrchy s otvory. Vypouklé povrchy odrážejí více záření do prostoru, čímž se také snižuje intenzita záření detekována senzorem. Intenzita odraženého detekovaného záření je vyšší, když povrch má světlou lesklou barvu a je celistvý, neboli bez otvorů.

Třetím faktorem je vzdálenost mobilního robotu a tím i senzoru od povrchu odrazu diskutovaného záření. Při nárazu do překážky v případě nulové vzdálenosti je odražené záření blokováno nárazníkem, tím pádem robot překážku nedetekuje. Při poloze robotu v pásmu detekce překážky není překážka detekována při každém měření. Toto je způsobeno disipaci záření v prostoru, což v podstatě znamená, že intenzita snímaného záření nepostačuje na to, aby ji senzor vyhodnotil jako přítomnost překážky. Tento efekt lze využít k určení přibližné vzdálenosti robotu vůči překážce počítáním počtu měření, kdy je překážka detekována v určitém časovém intervalu, neboli vůči určitému volenému počtu měření.

3.1.2 Měření senzorem QRD1114

Tři senzory intenzity odrazu IR záření s analogovým výstupem jsou umístěné uprostřed spodní přední části mobilního robotu *Obr. 19*. Způsob měření je podobný jako u senzoru SFH5110 s tím rozdílem, že *IR−LED* dioda je už zabudovaná do těla senzoru. Snímačem v senzoru je fototranzistor typu *NPN*.



Obr. 19 – umístění senzorů QRD1114 (pohled zdola)

Tyto senzory poskytují analogovu hodnotu, která je následně pomocí 8bitového *ADC* mikrokontroléru PICAXE-28X1 převedena na hodnotu o rozsahu 0÷255. Vyšší hodnoty zmíněného rozsahu jsou určeny pro tmavší a matnější barvy povrchu, od kterého se záření odráží. Dioda je zakomponována z důvodu zlepšení rozlišitelnosti, senzor díky konstrukčnímu řešení mobilního robotu je rušen minimálně, což znamená minimální vliv okolní osvětlení na měření robotu. Senzor obsahuje filtr denního světla.

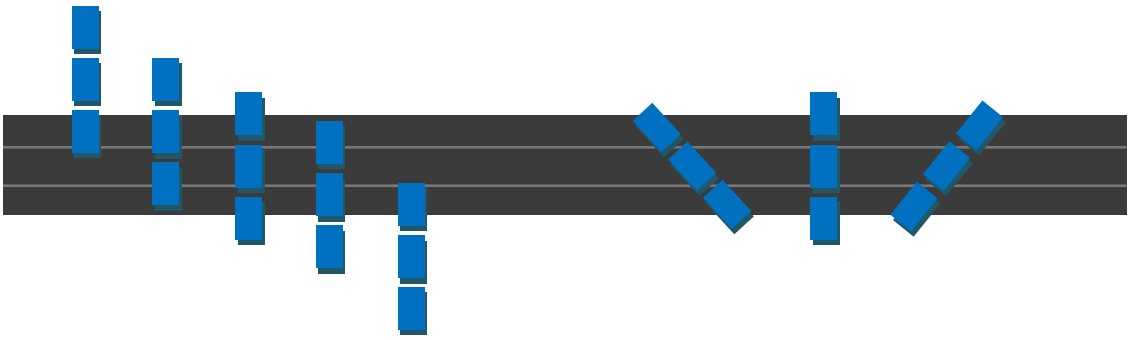
Jestli by vznikl požadavek na odstranění rušení je možné tento požadavek splnit. Postačí provést měření bez zapnutí *IR−LED* diody senzoru a následně provést měření se zapnutou *IR−LED* diodou senzoru. Následný rozdíl hodnot prvního a druhého měření bude mít za výsledek hodnotu rušení měření senzoru okolním osvětlením.

Problémem pro tento typ měření může představovat rychlá změna polohy mobilního robotu během měření, když se mění vlastnosti snímaného povrchu, jelikož ADC potřebuje určitou dobu na převod hodnoty poskytnuté senzorem QRD1114. Toto se dá řešit zvýšením kmitočtu vnitřního oscilátoru kontroléru PICAXE-28X1, a tím i zrychlením převodu ADC, zároveň se zvýšením kmitočtu vnitřního oscilátoru však zvýší energetická spotřeba kontroléru.

Pro účely detekce čáry a následné možnosti přesnějšího určení potřebné reakce mobilního robotu byla čára rozdělená na tři pásma, čímž byly pro jeden senzor v návaznosti na předchozí měření určeny tři uvažované polohy vůči čáře. Polohy jsou mimo čáru, nad okrajem čáry a nad středem čáry.

Budou-li brány v potaz použití tří senzoru QRD1114 a předchozí měření, robot si bude pamatovat předchozí polohy vůči čáře, rozšíří se možný počet poloh vůči čáře na pět s možností určení potřebné změny směru pohybu mobilního robotu za účelem sledování čáry. *Obr. 20*

Rozdělením čáry na více pásem a volba více možných poloh senzorů vůči čáře dovolí přesnější určení polohy mobilního robotu vůči čáře. Blok rozhodování programu by se stal úměrně složitější.



Obr. 20 – uvažované polohy senzorů QRD1114 vůči sledované čáře rozdělené na tři pásma pomyslnými světlými liniemi pro účely rozhodování, a) na sebe navazující posloupnost poloh v čase, b) příklad možné polohy senzorů se stejným vyhodnocením (stejné intervaly hodnot)

3.1.3 Zhodnocení způsobu měření

Za předpokladu že řídicí kontrolér, jak je tomu v tomto případě, řeší iniciací a čtení výsledků měření ze senzorů, pak při programování se musí brát v potaz doba trvání měření, jelikož zavádí do vykonávání programu určité zpoždění a tím i zpoždění reakce mobilního robotu na vnější podněty. Pro řešení určitých aplikací je však nezbytné měřit několikrát s prodlevou po sobě s ukládáním informací do paměti mobilního robotu a to z důvodu umožnění vyhodnocování získaných informací a získaní přesnějšího povědomí o potřebné reakci mobilního robotu na okolí.

V případě dostupné velké paměti pro ukládání dat u mobilního robotu lze zavést do programu řídicího kontroléru funkce učení a tím i zefektivnění plnění požadavků aplikací, jak je tomu u nadřazených úrovní výpočetní techniky. Dalším způsobem zefektivnění potřebných reakcí mobilního robotu je matematická simulace na základě známých informací a předpokladů vycházejících ze znalosti podmínek pro plnění aplikace. Následné testování v reálných podmínkách umožní ověření výpočtu a předpokladů užitých v matematické simulaci. Výsledkem bude znalost závislosti, které budou zavedeny do programu mobilního robotu, kde budou sloužit jako jádro pro rozhodování o jeho reakcích. Cílem programátora pak je omezit prodlevy způsobené měřením na co nejmenší možné, což jako nejlepší řešení činí ne co nejpřesnější měření podpořené velkým množstvím přeměřování za předpokladu klidu mobilního robotu, ale omezený počet měření postačující k určení základních informací a potřebných akcí neboli přibližného nastínění situace pro mobilní robot.

3.2 Aplikace sledování čáry

Úkolem programu mobilního robotu pro tuto aplikaci je řešit sledování čáry, tak aby aspoň jeden senzor byl vždy nad čarou. Požadavkem na program bylo, aby robot nejezdil po čáře v trhavých pohybech, neboli na určitou plynulost pohybu.

3.2.1 Předpoklady pro aplikaci

Předpokladem pro programování mobilního robotu UMU28A je znalost programovacího jazyka *Basic*, programovací prostředí PICAXE obsahuje manuál se seznamem příkazů a jejich popisem, znalost typu a připojení použitých akčních, řídicích a senzorických prvků. Při tvorbě programu je zapotřebí brát v úvahu okolní podmínky, které budou působit na UMU28A. Co konkrétně bude pro programování důležité určeno typem použitých prvků v mobilním robotu a jejích odezvy. Pro práci na zadané úloze bylo poskytnuto prostředí ***Obr. 21***, ve kterém má být výsledný program dimenzován.

Obr. 21 – dráha pro aplikaci "sledování čáry"

Délka prostředí činí 1,5m, šířka činí 75cm, výška bočního ohrazení činí 5cm. Po zhlédnutí prostředí ***Obr. 21***, na které se má program robotu dimenzovat, je možné vyvodit kritéria pro řízení, jsou to barva čáry, v daném případě černá, a šířka čáry, která činí 1cm.

Předpoklady zavedené pro psaní programu jsou tudíž *černá barva čáry* a dráha touto čarou vyznačená *bez ostrých zatáček*. Bylo třeba ověřit, zda lesklý povrch nebude *rušit* senzory QRD1114 při měření. Při *nadměrných hodnotách osvětlení* v místnosti, ve které se nacházela uvažovaná dráha, by k ovlivnění výsledku měření mohlo dojít. Takže se zavedl předpoklad o přesnosti hodnot měření.

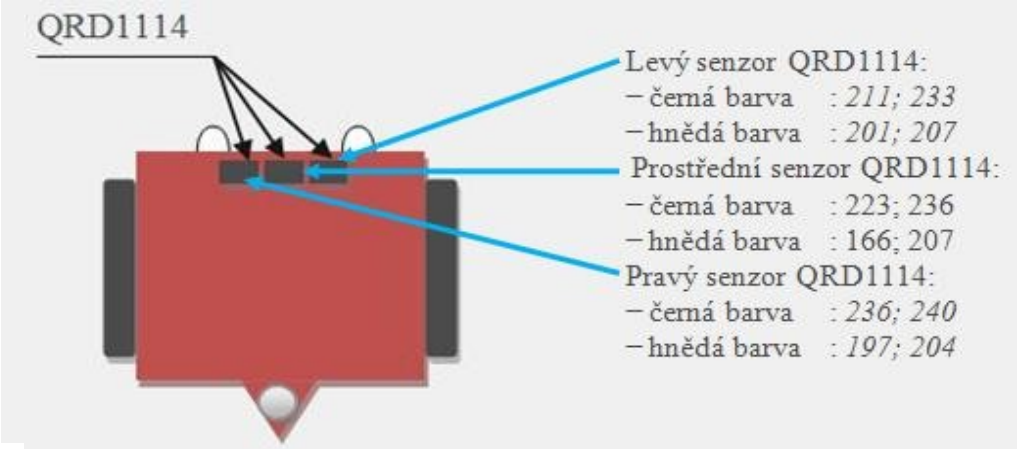
Zavedené předpoklady byly při psaní programu pro aplikaci sledování čáry řešeny a ověřovány zkoušením. Měřením byly zjištěny intervaly hodnot potřebné pro část programu pro rozhodování o reakci robotu. Plynulost pohybu byla řešená dvouúrovňovým rozhodováním robotu. Měřením bylo zjištěno, že rušení v cílové místnosti bylo minimální, avšak pro předcházení možných problému byly intervaly hodnot pro rozhodování robotu zvětšeny.

V aplikaci pro sledování čáry se užije tří senzorů typu QRD1114 (šířka jednoho senzoru činí 3 mm) umístěných podél předního nárazníku v 1 mm vzdálenosti od sebe (jsou situovány ve uprostřed spodní části nárazníku). Jedná se o senzory obsahující LED diodu a fototranzistor, tudíž senzor poskytuje analogový signál, který se za pomocí *ADC* (8 bitový) implementovaného v kontroléru PICAXE převede na číselnou hodnotu (0-255).

Obr. 22 – použité senzory použité pro detekci čáry

Princip měření spočívá v tom, že záření (směrované) působené diodou se odráží od povrchu a je zachyceno fototranzistorem. Síla odraženého záření se liší v závislosti na lesklosti (popř. matnosti) a barvě povrchu (materiálu), od kterého se záření odráží, tzn.: že je možné rozlišit barvy pomocí tohoto senzoru.

Pro každý ze senzorů QRD1114 v každé ze čtyř hlavních uvažovaných poloh (*Obr. 20*) při plynulém pohybu byly zjištěny tyto orientační hraniční hodnoty:



Obr. 23 – rozložení senzorů QRD1114 (pohled zdola), zjištěné hraniční hodnoty pro příslušné senzory

Jako akční člen mobilnímu robotu, v této i další aplikaci, slouží dva stejnosměrné motory s převodem. Motory ovládá procesor 16F688, který se pro nás chová jako "černá skříňka" ovládána přes sériovou sběrnici (jedno-bajtovou proměnnou). Rozsahy hodnot pro ovládání motorů mobilního robotu:

Pro **levý** motor – *62 → 1* (zrychlování - vzad); *64 → 127* (zrychlování - vpřed)
Pro **pravý** motor – *190 → 129* (zrychlování - vzad); *192 → 254* (zrychlování - vpřed)
Pro **levý** motor – *62 → 1* (zrychlování - vzad); *64 → 127* (zrychlování - vpřed)
Pro **pravý** motor – 128*; 191* (stop)
Pro **oba** motory – *0; 63* (stop)

3.2.2 Program pro zjištění potřebných hodnot

Pro přesné zjištění hodnot potřebných pro rozhodování je možné provést měření a zobrazit výsledek na obrazovce PC propojeným s mobilním robotem sériovou linkou. Výsledky měření se budou vypisovat v okně *Terminal* ve vývojovém prostředí *PICAXE Programming Editor Obr. 14*. Program pro provedení takovéhoto měření by vypadal následujícím způsobem:

```
#picaxe          28x1    ; Nastaveni direktivy (definice instrukcí pro kompilátor o cílovém
                                 ; kontroléru).
symbol SL =      1       ; Přiražení symbolu VÝSTUPU (OUT1) pro spodní LED (zlepší
                                 ; rozeznávání barev).
symbol LB =      2       ; Přiražení symbolu VSTUPU (ADC2) pro spodní levý fototranzistor
                                 ; (převedená hodnota).
symbol MB =      0       ; Přiraženi symbolu VSTUPU (ADC0) pro spodní prostřední
                                 ; fototranzistor (převedená hodnota).
symbol RB =      1       ; Přiraženi symbolu VSTUPU (ADC1) pro spodní pravý fototranzistor
                                 ; (převedená hodnota).
symbol LBP=      B0      ; Přiraženi paměti pro odměřenou hodnotu z levého spodního
                                 ; senzoru v čase t.
symbol MBP=      B1      ; Přiraženi paměti pro odměřenou hodnotu z prostředního
                                 ; spodního senzoru v čase t.
symbol RBP=      B2      ; Přiraženi paměti pro odměřenou hodnotu z pravého spodního
                                 ; senzoru v čase t.

Main:
                         ; Návěští programu.
        high             SL
                         ; Zapnutí spodních IR−LED.
        readadc          LB, LBP
                         ; Čtení a zápis do paměti z levého spodního fototranzistoru.
        readadc          MB, MBP
                         ; Čteni a zápis do paměti z prostředního spodního fototranzistoru.
        readadc          RB, RBP
                         ; Čteni a zápis do paměti z pravého spodního fototranzistoru.
        low              SL
                         ; Vypnuti spodní IR−LED za účelem šetřeni energie.
        sertxd   ("Levy dolni senzor       : ",#b0,13,10)
                         ; Poslání výsledků čteni do PC.
        sertxd   ("Prostredni dolni senzor: ",#b1,13,10)
                         ; Poslání výsledků čteni do PC.
        sertxd   ("Pravy dolni senzot      : ",#b2,13,10)
                         ; Poslání výsledků čteni do PC.
        pause            500
                         ; Časová prodleva půl sekundy před dalším měřením a jeho zobrazením,
                                 ; aby programátor stihl odečíst hodnoty měření
        goto             Main
                         ; Skok na začátek programu pro opětovné měřeni a zobrazeni hodnot
                                 ; (zajištění cyklického vykonávání programu).
end                      ; Ukončení programu.
```

Pro tuto aplikaci má čára černou barvu (interval hodnot *230-255*) a zbývající povrch lesklou hnědou barvu (na světlejších barvách náleží do hodnot menších než *221*).

3.2.3 Návrh podprogramů

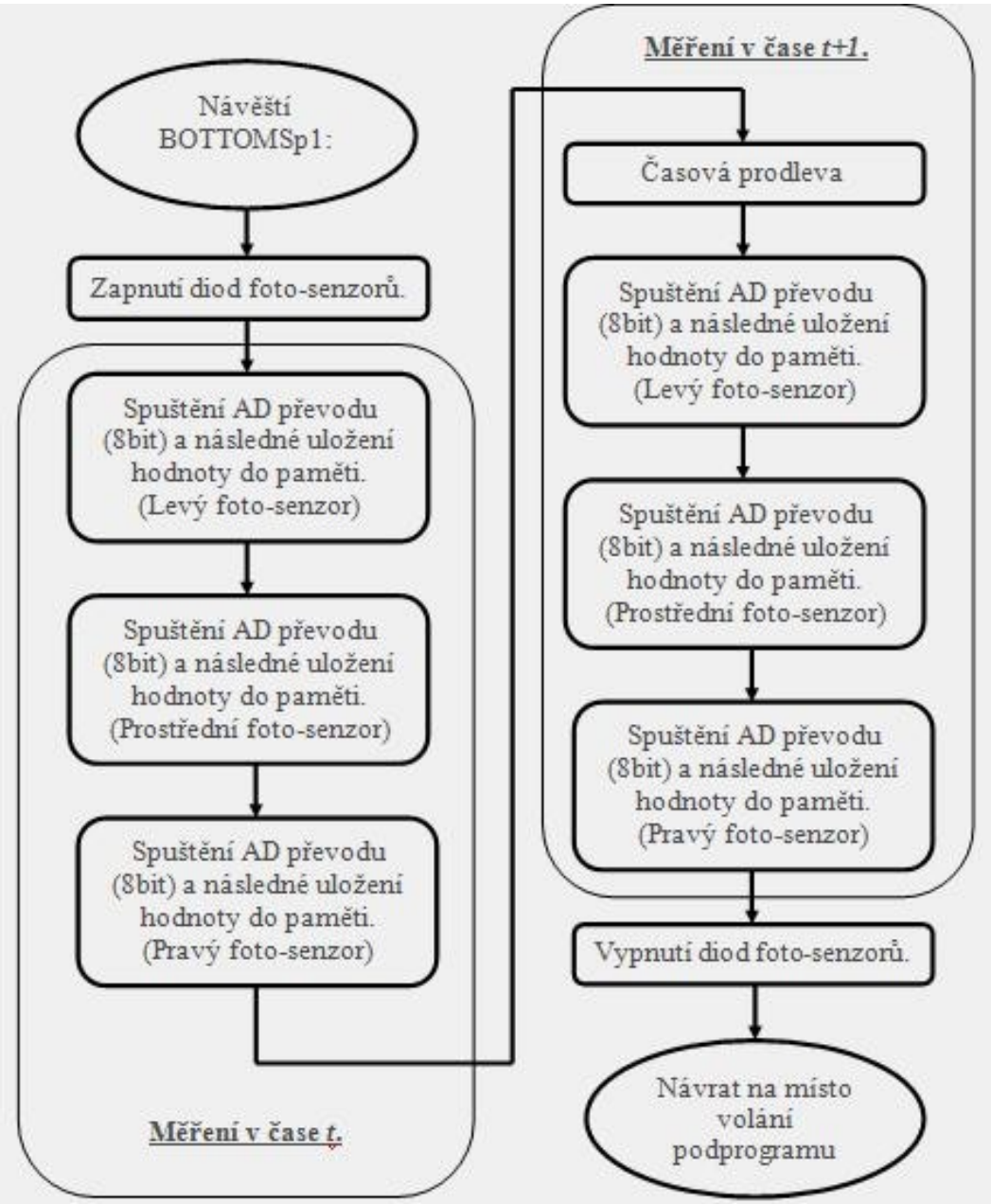
Nejdříve se definují podprogramy pro cyklicky vykonávané činnosti jako odečítaní hodnot poskytovaných senzory a posílaní řídicích instrukcí motorům, je tak činěno z důvodu šetření pamětí pro program a přehlednost (pochopitelnost) programu. Výhodou je, že pro každou možnost rozhodnutí programu není nutno znovu vypisovat příkazy pro vykonávání činnosti, postačuje uložení příslušných hodnot pro řízení motorů do paměti, odkud jsou v příslušný čas vyzvednuty a poslány sériovou linkou kontroléru ovládajícímu motory. Pro vykonání činnosti potom postačí jen volání nadefinovaného podprogramu.

Obr. 24 – blokové schéma podprogramu pro řízení motorů mobilního robotu UMU28A

Řízení akčních členů neboli motorů není přímé řízení, jelikož motory jsou připojeny k jinému mikrokontroléru, který s PICAXE-28X1 komunikuje pomocí sériové linky. K řízení se užívá přenos 8bitové proměnné. Každému akčnímu členu náleží interval hodnot z 8bitového rozsahu přenášené informace. Pro řízení obou motoru se musí pro každý motor odeslat příslušná řídicí hodnota ze zmíněného rozsahu (strana 24).

Poslat řídicí hodnoty je možné příkazem *serout*, jehož popis je možné najít přímo ve vývojovém prostředí v druhém manuálu pro PICAXE v záložce *Help*. Použitá přenosová rychlost bude 2400 bitů za sekundu, vyslání hodnot se provede pomocí sedmého pinu kontroléru PICAXE-28X1.

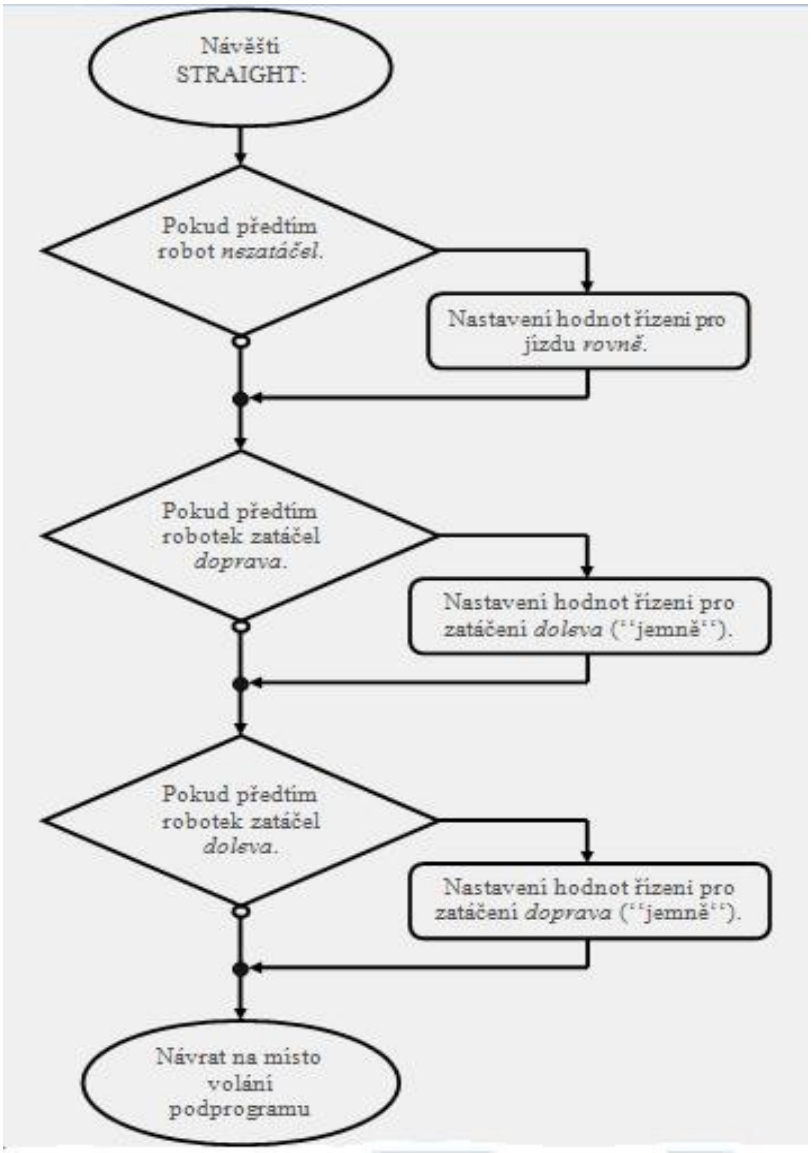
Pro volání podprogramu se použije příkaz *gosub*. Pro návrat na řádek programu volající podprogram se užije příkaz *return*. Celistvý seznam příkazu pro mikrokontroléry PICAXE je možno nalézt na stránkách výrobce [5].



Obr. 25 − blokové schéma podprogramu detekci čáry spodními senzory QRD1114 mobilního robotu UMU28A

Způsob měření senzory QRD1114 byl vysvětlen v příslušné kapitole 3.1.2. Pro iniciaci analogově-digitálního převodu se užije příkaz *readadc*. K zapnutí a vypnutí IR diod foto-senzorů byly užity příkazy *high* a *low*. Celistvý seznam příkazu pro mikrokontroléry PICAXE je možno nalézt na stránkách výrobce [5].

Použité piny kontroléru pro měření, nebo li analogově-digitální převod, jsou 2., 3. a 4. (ADC 0÷2). Měření je provedeno pro čas *t* a *t+1*, což jenom naznačuje použití volené doby prodlevy mezi měřeními. Takto je měření řešeno pro možnost užití souvislosti akce-reakce při řízení, tzn. robot je schopen vyhodnotit, zda sleduje čaru a jestli a jak moc z ní vybočuje *Obr. 20* (a).

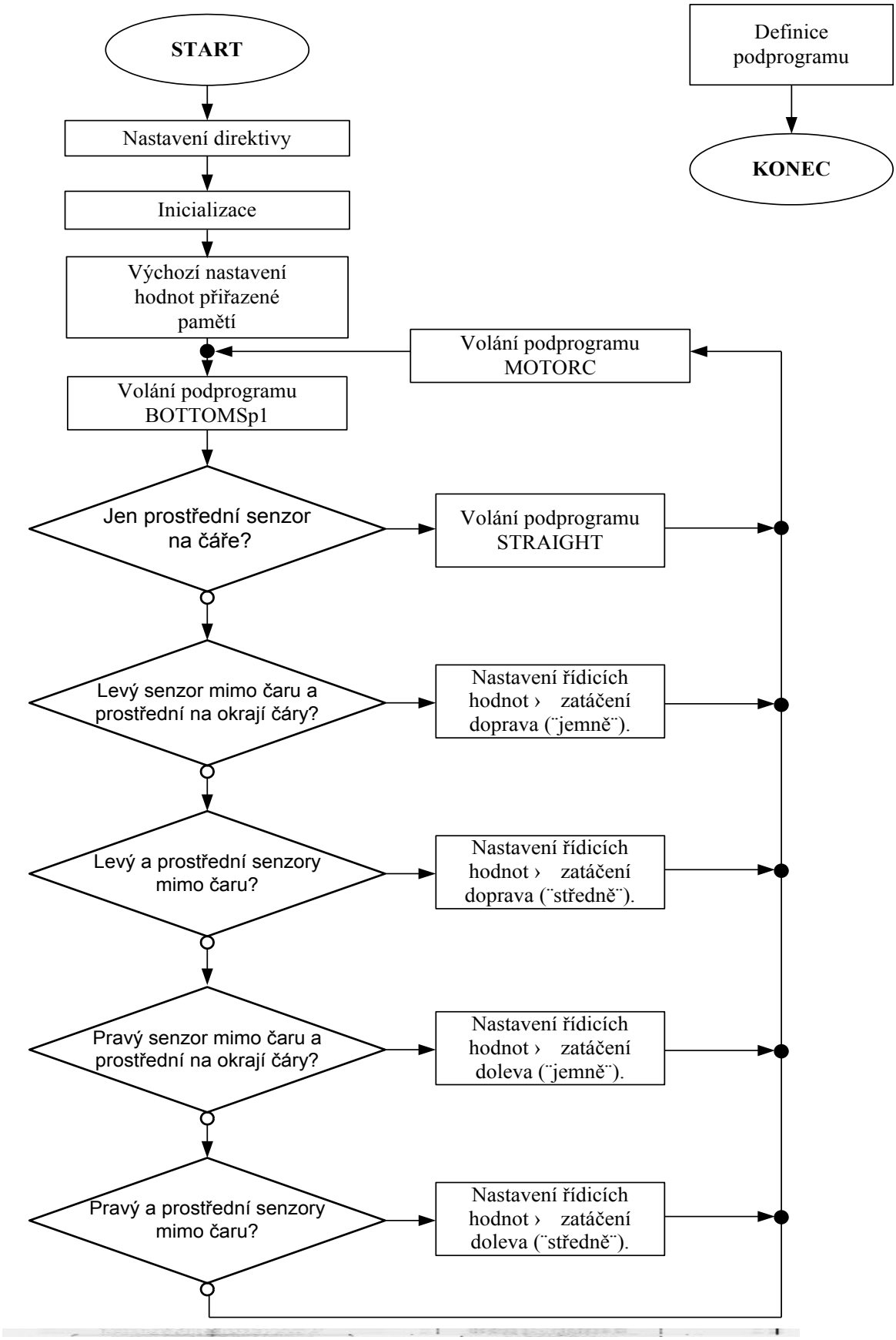


Obr. 26 − blokové schéma podprogramu pro ověření směru jízdy mobilního robotu UMU28A

Tento podprogram funguje jako další úroveň rozhodování v podmíněném větvení, když jsou splněny podmínky pro toto větvení ***Obr. 20*** (b). Kontrolér si hlídá, zda robot jel předtím přímo po čáře nebo po zatáčení najel na čaru uvažovaným způsobem přičemž způsob měření dovoluje určit posloupnost děje.

Užití tohoto podprogramu je podmíněno požadavkem na plynulost pohybu řízeného robotu, aby trajektorie pohybu nepřipomínala pilovitý průběh. Byly použity příkazy pro podmíněné větvení programu *if…then* a *elseif*. Celistvý seznam příkazu pro mikrokontroléry PICAXE je možno nalézt na stránkách výrobce [5].

3.2.4 Návrh programu



Obr. 27 − blokové schéma programu pro sledování čáry

Nyní lze psát program, jelikož možná zjednodušení byla provedena. Pro volání podprogramu (tzv. vnoření) je použit příkaz *gosub*. Je možno provést maximálně 255 vnoření bez užití příkazu k návratu na řádek volání příslušného podprogramu (*return*).

Na začátku programu se píše direktiva pro požadovaný mikrokontrolér řady PICAXE, jedná se o instrukce pro kompilátor zpracovávající program v Basicu pro dotyčný mikrokontrolér při nahrávání programu do kontroléru.

Po startu programu, nastavení direktivy, proběhne inicializace neboli základní přiřazení v programu vyžitých názvu pro paměťové bajty případně vstupní či výstupní piny kontroléru.

Po inicializaci je nastavení výchozích hodnot paměťových proměnných, jelikož může dojit k tomu, že v paměti zůstanou uložené proměnné z předchozího programu či motory kvůli předchozímu programu stále poběží z toho důvodu, že jim nebyl dán pokyn k zastavení, čemuž předchází tento blok.

Následuje cyklicky vykonávaný blok řízení. *Prvním* krokem v tomto bloku je nejprve volání podprogramu pro měření spodními foto-senzory QRD1114. *Druhým* krokem je samotné rozhodování neboli podmíněné větvení programu. Počet větvení definuje uvažované možnosti, které mohou nastat s přihlédnutím k potřebné přesnosti snímaných veličin a reakci. Tato část programu v podstatě představuje umělou inteligenci mobilního robotu. Konkrétně pro tento případ je uvažováno pět možných stavu a tím i pět možných akcí vykonaných kontrolérem jsou-li podmínky toho či jiného bloku splněny přičemž podmínky byly v tomto programu stanoveny tak, že jenom jednomu z bloku bude vyhověno. Definovaných poloh samozřejmě může být více i aplikované s větší přesnosti musely by ale být stanoveny pevné podmínky a tím i předpoklady pro psaní programu. Pro tento případ by to byly dobrá úroveň osvětlení trati a pomalý pohyb mobilního robotu, jelikož řízení motorů má jistou prodlevu způsobenou sériovou komunikací. *Třetím* krokem je volání podprogramu pro řízení motorů neboli sériovou komunikaci s kontrolérem PIC 16F688 řídicím motory. Není-li splněna ani jedna podmínka podmíněného větvení je odeslán motorům pokyn z předchozího cyklu bloku řízení. Případ kdy, všechny foto-senzory robotu nejsou nad čárou, není ošetřen, aneb hledání čáry není ošetřeno.

Po dokončení cyklu měření, rozhodnutí a reakce se provede skok na návěští, neboli místo programu označující určité místo, v tomto případě začátek cyklu.

Za hlavním blokem měření, rozhodování a reakce se nachází část programu, kde jsou definovány podprogramy pro program řešící danou aplikaci. Činí se tak z důvodu šetření místa programu, aby se při každém měření nebo větvení nemusela psát celá procedura řešící požadovanou činnost. Po podprogramech je ukončení textu programu.

3.2.5 Zhodnocení navrženého programu

Po napsání byl program testován pro zadané prostředí *Obr. 21*. Pozorováním při testování programu byly zjištěny a následně opraveny chyby v programu. Program úspěšně řeší požadovanou aplikaci sledování čáry. Řešení je navrženo s ohledem na jednoduchost a přehlednost. Způsob řízení robotu však je hrubý s omezenou přesnosti, toto by se dalo řešit zavedením matematických výpočtu pro regulaci do bloku rozhodování v programu, avšak nebylo tak učiněno z důvodu potřeby nároku na nepostačující paměť pro proměnné kontroléru PICAXE-28X1. Pro zlepšení regulovatelnosti je možné užitím příkazu *freq* zvýšit frekvenci vnitřního oscilátoru kontroléru, čímž by se zrychlily procesy a také zvětšila energetická spotřeba kontroléru.

3.3 Aplikace minisumo

Úkolem programu mobilního robotu pro tuto aplikaci je řešit zápas v minisumo**,** aneb vytlačení soupeřícího mobilního robotu z arény. Požadavkem bylo dodržování uvedených pravidel [12] a jednoduchost a názornost programu.

3.3.1 Předpoklady pro aplikaci

Jak již bylo uvedeno v kapitole 3.2.1, předpokladem pro programování mobilního robotu UMU28A je znalost programovacího jazyka *Basic*, programovací prostředí PICAXE obsahuje manuál se seznamem příkazu a jejich popisem, znalost typu a připojení použitých akčních, řídicích a senzorických prvků. Při tvorbě programu je zapotřebí brát v úvahu okolní podmínky, které budou působit na UMU28A. Co konkrétně bude pro programování důležité je určeno typem použitých prvků v mobilním robotu a jejich odezvami. Pro práci na zadané úloze bylo vyrobeno prostředí *Obr. 28*, dle specifikovaných pravidel 40[12], na které má být výsledný program dimenzován.

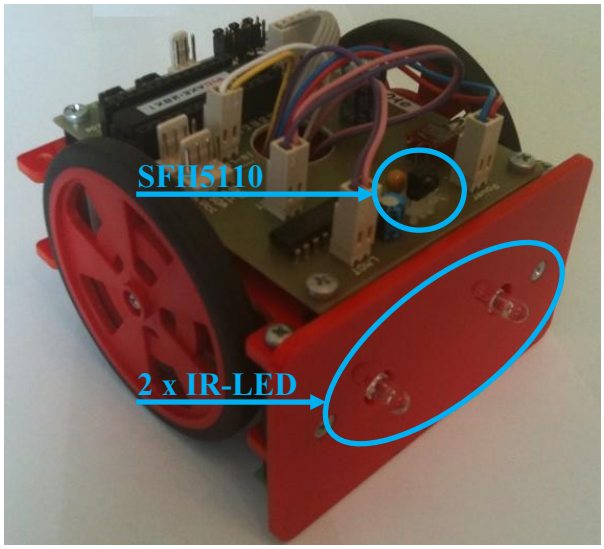
Obr. 28 − aréna pro aplikaci "minisumo"

Po zhlédnutí prostředí ***Obr. 28***, ***Obr. 21***, na které se má program robotu dimenzovat, je možné vyvodit kritéria pro řízení, jsou to barva povrchu a ohraničení, v daném případě černá a bílá, a odrazivost ohraničujícího bílého okraje arény a povrchu soupeřícího mobilního robotu.

Předpoklady zavedené pro psaní programu jsou tudíž *černá matná barva povrchu arény* a *bílá lesklá barva okraje arény*. Co se tyče odrazivosti, okraje arény se nebudou interpretovány jako překážka.

Zavedené předpoklady byly při psaní programu pro aplikaci minisumo řešeny a ověřovány zkoušením. Měřením byly měřeny intervaly hodnot potřebné pro část programu pro rozhodování o reakci robotu. Dynamičnost pohybu vnáší určitou setrvačnost mobilního robotu, konkrétně analogově-digitální převod hodnot probíhající při ověřování najetí na okraj arény nestíhá zpracovat přechod z jedné barvy na druhou a tím vzniká možnost samo-vyřazení mobilního robotu ze zápasu, z toho důvodu byly rozšířeny okraje arény. Následným zkoušením bylo zjištěno, že robot není schopen reagovat na okraje arény, jelikož doba měření senzorem SFH5110 je příliš dlouhá. Počet měření a tím i celková doba měření senzorem SFH5110 byla zkrácena.

V aplikaci minisumo se užijí oba druhy senzorů a to typu QRD1114 pro detekci okraje arény (*Obr. 30*, *Obr. 21*) a SFH5110 pro detekci soupeřícího mobilního robotu (*Obr. 29*). Způsoby měření těmito senzory byly uvedeny v kapitolách 3.1.1 a 3.1.2.



Obr. 29 – senzor SFH 5110 užitý k detekci soupeře a 2 IR-LED diody

Obr. 30 – senzory typu QRD1114 užité k detekci okraje arény

Intervaly měřených hodnot byly navrženy tak, aby informace ze senzorů QRD1114 byly vyhodnocovány jako dvoustavové a ze senzoru SFH5110 jako třístavové. Navýšení přesnosti měření, zvýšením počtu měření, by působilo větší setrvačnost, jelikož samotný děj měření zabírá určitou dobu a při zvětšením počtu měření se úměrně zvětšuje prodleva před reakcí mobilního robotu, proto se musí volit optimální počet měření.

Jako akční člen mobilnímu robotu slouží dva stejnosměrné motory s převodem. Motory ovládá procesor 16F688, který se pro nás chová jako "černá skříňka" ovládáná přes sériovou sběrnici (jedno-bajtovou proměnnou). Rozsahy hodnot pro ovládání motorů robotu:

Pro **levý** motor – *62 → 1* (zrychlování - vzad); *64 → 127* (zrychlování - vpřed)
Pro **pravý** motor – *190 → 129* (zrychlování - vzad); *192 → 254* (zrychlování - vpřed)
Pro **levý** motor – *62 → 1* (zrychlování - vzad); *64 → 127* (zrychlování - vpřed)
Pro **pravý** motor – 128*; 191* (stop)
Pro **oba** motory – *0; 63* (stop)

3.3.2 Program pro zjištění potřebných hodnot

Pro přesné zjištění hodnot potřebných pro rozhodování je možné provést měření a zobrazit výsledek na obrazovce PC propojeným s mobilním robotem sériovou linkou. Výsledky měření se budou vypisovat v okně *Terminal* ve vývojovém prostředí *PICAXE Programming Editor Obr. 14*. Program je řešen obdobně, jako v kapitole 3.2.2, pro sériovou komunikaci. Fragment programu řešícího měření odezvy z jedné IR-LED by vypadal následujícím způsobem:

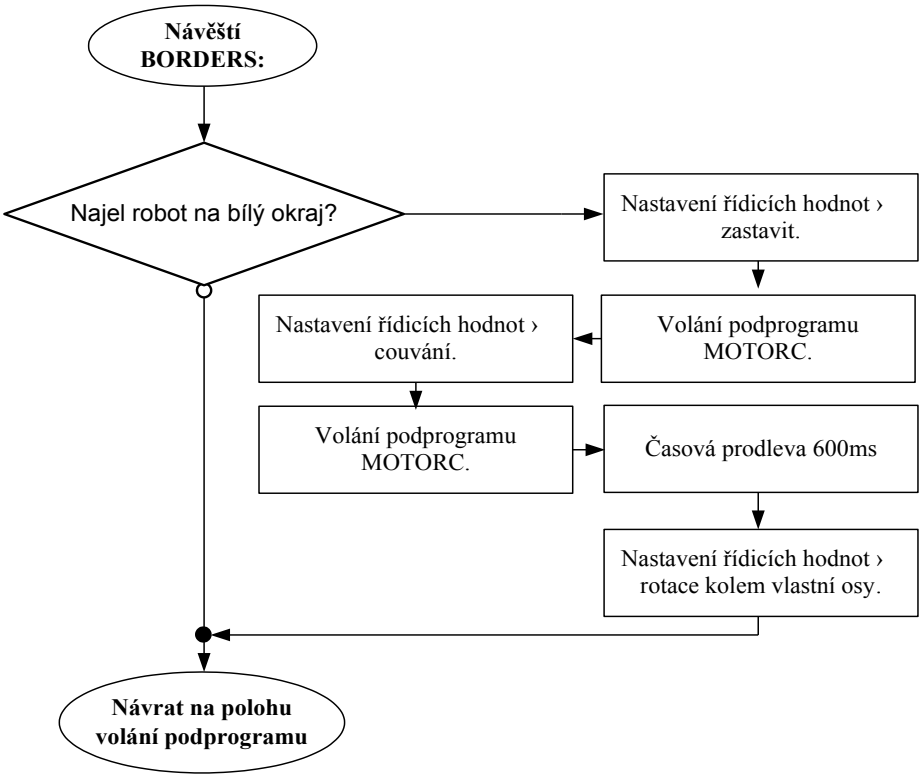
```
; Přiřazení pinů a paměťových proměnných:
symbol  IRS  = pin5     ; Definice VSTUPU IR-senzoru (IN5 -> SFH5110).
symbol  PWMV = 1        ; Definice VÝSTUPU pro PWM (PWM1 -> port C).
symbol  PWMP = 2        ; Definice VÝSTUPU pro, stanovení pravé modulované IR-LED
                                ;(OUT2).
symbol LCNT   = B0      ; Přiřazení pamětí pro, čítaní odražených pulsů z levé strany
                                ; (Left  Counter)
symbol RCNT   = B1      ; Přiřazení pamětí pro, čítaní odražených pulsů z pravé strany
                                ; (Right Counter)
symbol MCNT= B2         ; Přiřazení pamětí pro, čítaní počtu měření senzorem SFH5110.

Fragment programu:
        LPIRL:
                        ; Nevěští měřeni odezvy (odrazu) z levé IR-LED.
                low             PWMP
                        ; Nastavení log 0 na výstup OUT2 pro, umožnění PWM regulace
                                ; levé IR LED (viz schéma zapojení).
                pwmout          PWMV,26,26
                        ; Zapnutí PWM při f=38 kHz, plnění 25%.
                        ; (Duty Cycle - doba periody po kterou signál bude v log.1).
                pauseus         100
                        ; Doba měření IR-senzoru (SFH5110) je 1ms.
                if      IRS=1   then
                                LCNT=LCNT
                        ; Pokud IR-senzor posílá do procesoru log 1, pak se do paměti uloží
                                ; (hodnota +0) že není detekována žádna překážka,
                                else
                                LCNT=LCNT+1
                                ; v opačném případě (IR-senzor posila log 0) se do paměti uloží,
                                ; že je detekována překážka (hodnota +1).
                endif   ; Konec podmíněného větvení programu.
                pwmout          PWMV,255,0
                        ; Zastavení (vypnuti) PWM (Duty Cycle -> 0%).
                pause           9
                        ; Časová prodleva (9ms) - obnovení citlivosti IR-senzoru (SFH5110).
```

Pro tuto aplikaci má povrch arény černou barvu (interval hodnot *230÷255*) a okraj arény bílou lesklou barvu (světlejší barvy náleží do hodnot menších než *0÷229*). Pro stanovení přítomnosti překážky bylo voleno 13 opakování měření senzorem SFH5110, přítomnost překážky bude signalizována 13/13 detekcemi překážky, volný prostor 0/13 detekcemi překážky a <1÷12>/13 detekci překážky se interpretuje jako přítomnost soupeřícího robotu před robotem.

Plněním se rozumí kolik pulsů za periodu *PWM* se má realizovat, čímž se reguluje výkon vyzařovaný *IR-LED* diodami, což dovoluje nastavovat maximální dosah měření pomocí SFH5110.

3.3.3 Návrh podprogramů

Nejdříve se definují podprogramy pro cyklicky vykonávané činnosti jako odečítaní hodnot poskytovaných senzory a poslání řídicích instrukcí motorům. Je tak činěno z důvodu šetření pamětí pro program a přehlednost (pochopitelnost) programu. Výhodou je, že pro každou možnost rozhodnutí programu není nutno znovu vypisovat příkazy pro vykonávání činnosti, postačuje uložení příslušných hodnot pro řízení motorů do paměti, odkud jsou v příslušný čas vyzvednuty a poslány sériovou linkou kontroléru ovládajícímu motory. Pro vykonání činnosti potom postačí jen volání nadefinovaného podprogramu.



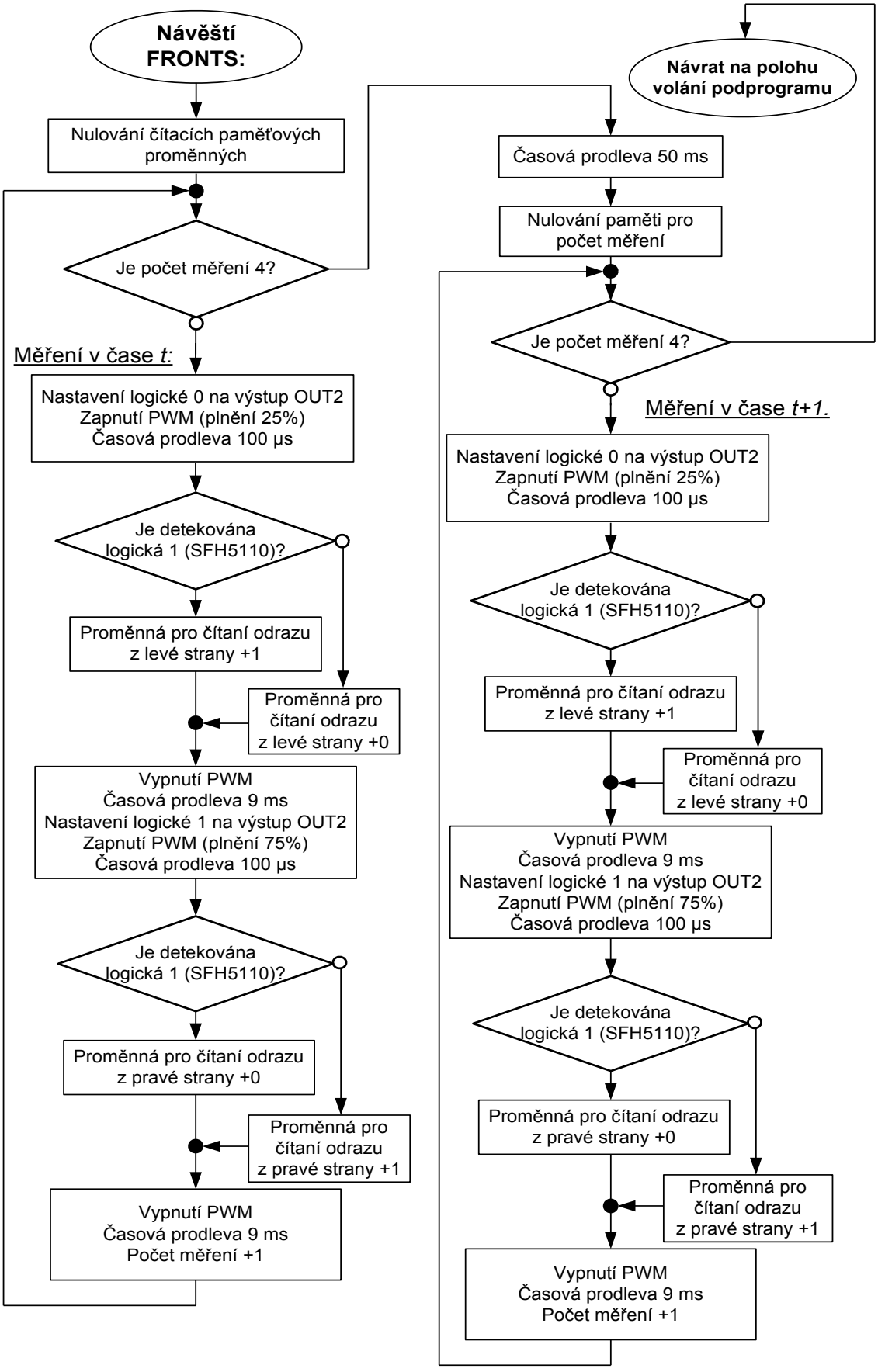
Obr. 31 − blokové schéma podprogramu pro detekci okraje arény

Reakce na detekci okraje arény je řešena posloupností předdefinovaných akcí z důvodu rozmístění senzoru a neschopnosti provést přesný analogově-digitální převod při pohybu mobilního robotu po aréně (detekce je řešená jako dvoustavová).

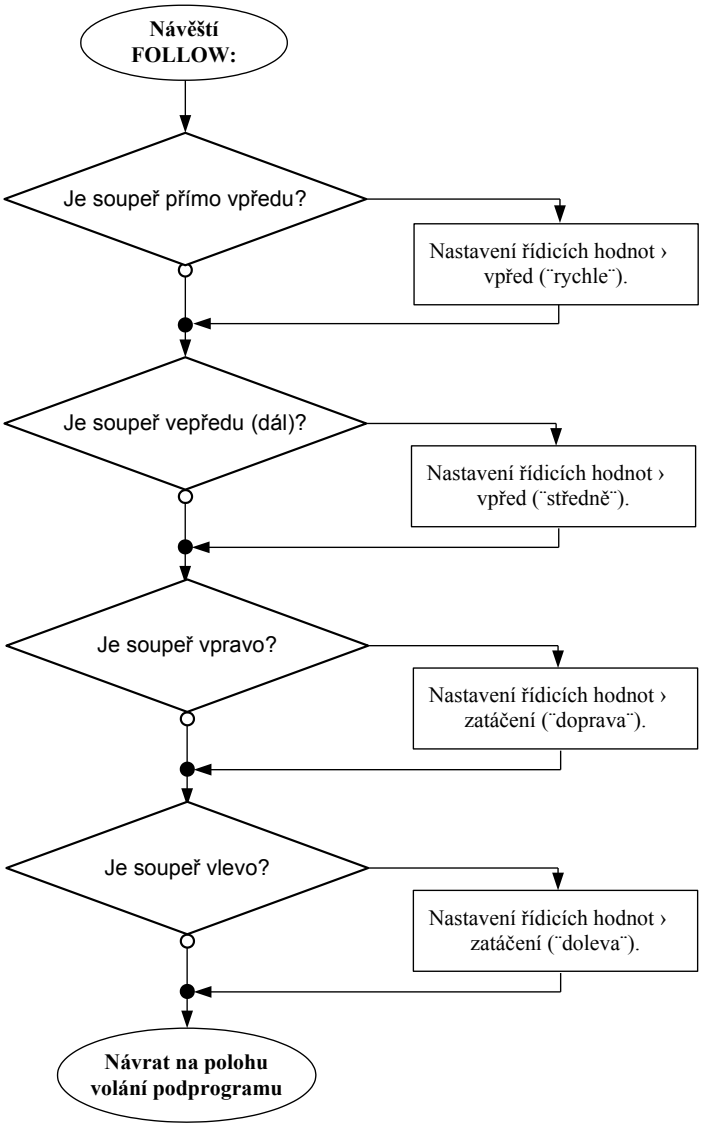
Popis měření a odůvodnění střídavého měření odrazu *IR* záření z levé a pravé IR-LED diod je popsáno v kapitole 3.1.1. Počet opakování měření byl volen s ohledem na vliv doby jednoho měření na prodlevu reakce motoru na podnět.

Pro měření byly užity příkazy *pwmout*, *for* a již výše zmíněné v předcházejících podprogramech. Celistvý seznam příkazu pro mikrokontroléry PICAXE je možno nalézt na stránkách výrobce [5].

Jelikož podprogramy pro měření pomocí senzorů QRD1114 a řízení motorů jsou už řešené v kapitole 3.2.3, nebudou již znova uváděny v této kapitole.



Obr. 32 – blokové schéma podprogramu detekce soupeřícího mobilního robotu senzorem SFH5110

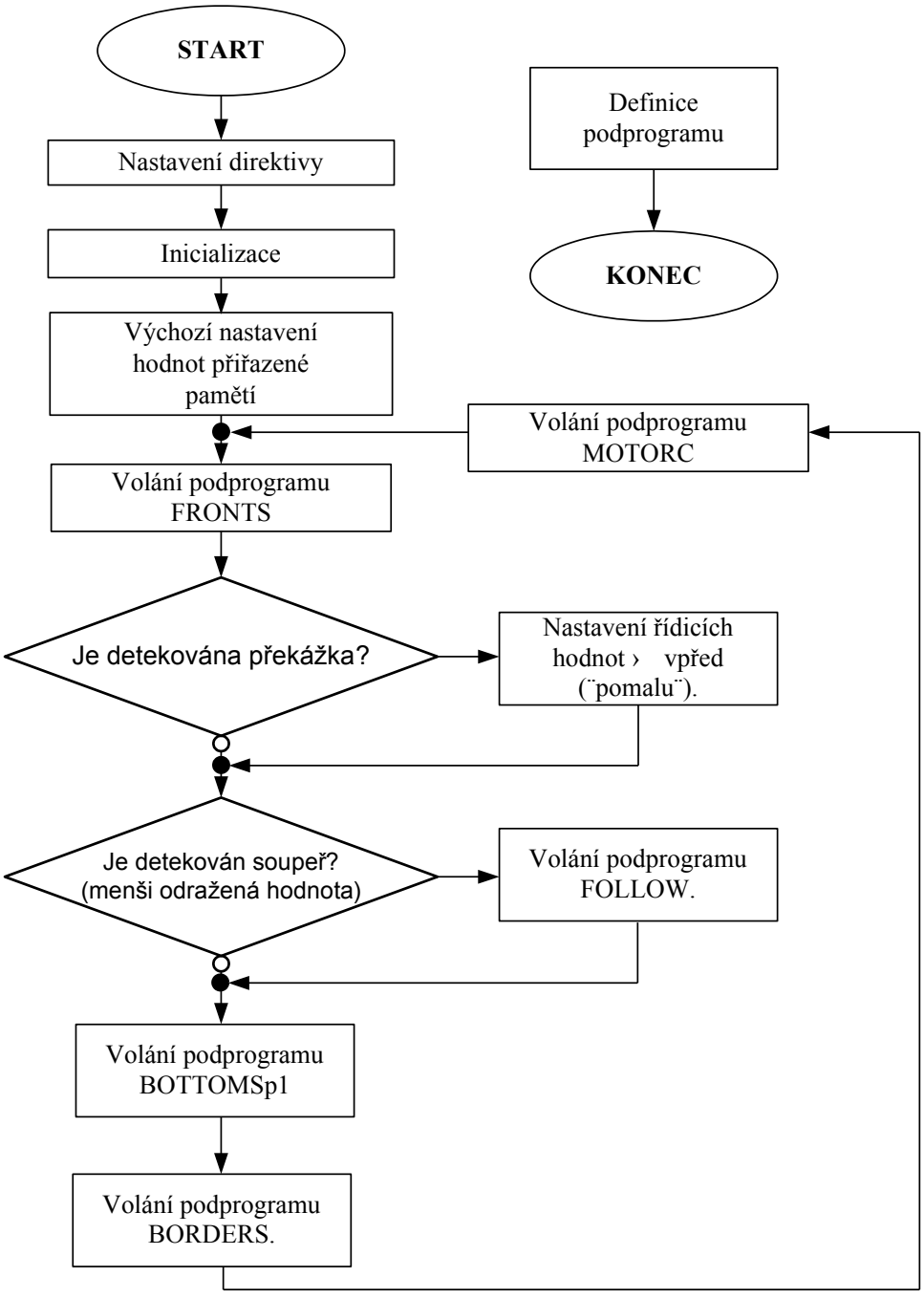


Obr. 33 – blokové schéma podprogramu pro sledování soupeře

Tento podprogram funguje jako další úroveň rozhodování v podmíněném větvení, když jsou splněny podmínky pro toto větvení. Kontrolér si hlídá, kde se soupeřící robot nacházel a kde se zrovna nachází, se stanoveným časovým odstupem a dle toho volí reakci na pohyb soupeřícího mobilního robotu.

Byly použity příkazy pro podmíněné větvení programu *if…then* a *elseif*. Celistvý seznam příkazu pro mikrokontroléry PICAXE je možno nalézt na stránkách výrobce [5].

3.3.4 Návrh programu



Obr. 34 − blokové schéma programu pro aplikaci ‘‘minisumo‘‘

Jelikož možná zjednodušení byla provedena, již už lze psát program. Pro volání podprogramu (tzv. vnoření) je použit příkaz *gosub*, je možno provést maximálně 255 vnoření bez užití příkazu k návratu na řádek volání příslušného podprogramu (*return*).

Na začátku programu se píše direktiva pro požadovaný mikrokontrolér řady PICAXE. Jedná se o instrukce pro kompilátor zpracovávající program v Basicu pro dotyčný mikrokontrolér při nahrávání programu do kontroléru.

Po startu programu, nastavení direktivy, proběhne inicializace neboli základní přiřazení v programu vyžitých názvu pro paměťové bajty případně vstupní čí výstupní piny kontroléru.

Po inicializaci dochází k nastavení výchozích hodnot paměťových proměnných, jelikož může dojit k tomu, že v paměti zůstanou uložené proměnné z předchozího programu, či motory kvůli předchozímu programu stále poběží z toho důvodu, že jim nebyl dán pokyn k zastavení, čemuž předchází tento blok. Následně program čeká určitou dobu, než je mobilní robot umístěn na startovní pozici, pak dá pokyn k jízdě přímo vpřed co nejrychleji, jelikož se předpokládá pozice soupeřícího mobilního robotu.

Následuje cyklicky vykonávaný blok řízení. *Prvním* krokem v tomto bloku je nejprve volání podprogramu pro měření senzorem SFH5110. *Druhým* krokem je samotné rozhodování neboli podmíněné větvení programu. Počet větvení definuje uvažované možnosti, které mohou nastat s přihlédnutím k potřebné přesnosti snímaných veličin a reakci. Tato část programu v podstatě představuje umělou inteligenci mobilního robotu. Konkrétně pro tento případ jsou uvažovány dva možné stavy a to detekuji soupeře nebo ne. Jsou-li podmínky toho či jiného bloku splněny přičemž podmínky byly v tomto programu stanoveny tak, že jenom jednomu z bloku bude vyhověno. Jestli senzor hlásí detekci překážky, znamená to, že robot detekuje přítomnost soupeře před mobilním robotem a tak volá program pro jeho sledování. *Třetím* krokem je volání podprogramu pro kontrolu najetí na okraj arény, dojde-li k takovému případu, robot provede sérii předdefinovaných akcí. *Čtvrtým* krokem je volání podprogramu pro řízení motorů neboli sériovou komunikaci s kontrolérem PIC 16F688 řídicím motory. Není-li splněna ani jedna podmínka podmíněného větvení je odeslán motorům pokyn z předchozího cyklu bloku řízení.

Po dokončení cyklu měření, rozhodnutí a reakce se provede skok na návěští, neboli místo programu označující určité místo, v tomto případě začátek cyklu.

Za hlavním blokem měření, rozhodování a reakce se nachází část programu, kde jsou definovány podprogramy pro program řešící danou aplikaci. Činí se tak z důvodu šetření místa programu, aby se při každém měření nebo větvení nemusela psát celá procedura řešící požadovanou činnost. Po podprogramech je ukončeni textu programu.

3.3.5 Zhodnocení navrženého programu

Po napsání byl program testován pro zadané prostředí *Obr. 28*. Pozorováním při testování programu byly zjištěny a následně opraveny chyby v programu. Program úspěšně řeší požadovanou aplikaci minisumo. Řešení je navrženo s ohledem na jednoduchost a přehlednost. Program řeší zápas dvou robotu přetlačováním. Tento program neobsahuje žádné zefektivnění schopnosti vytlačení soupeře z arény užitím *"manévrů"*. Po detekci okraje arény se robot snaží znovu do ní navrátit. Dojde-li k přejetí okraje arény robot, nedokáže rozpoznat, že prohrál, jelikož nemá dostačující paměť pro proměnné. Okraje arény musely být rozšířeny, jelikož robot nestíhal reagovat na informace poskytnuté senzory QRD1114. Pro zlepšení regulovatelnosti je možné užitím příkazu *freq* zvýšit frekvenci vnitřního oscilátoru kontroléru, čímž by se zrychlily procesy a také zvětšila energetická spotřeba kontroléru.

Závěr

Cílem zadání bakalářské práce bylo navrhnout výukové úlohy pro malý mobilní robot, včetně základního popisu robotu a vývojového prostředí.

Protože při programování použitého mikrokontroléru je potřeba také rozumět technickým prostředkům robotu, byla velká pozornost věnována popisu schématu zapojení součástí mobilního robotu a nadefinování (přiřazení) názvu pamětem a pinům kontroléru PICAXE28X1. Podrobně lze prostudovat příkazy a vlastnosti kontroléru v *datových listech (datasheet)*, k nimž je možné se dostat pomocí záložky *Help* v programovacím prostředí "*PICAXE Programming Editor*" nebo na internetových stránkách výrobců [1], [5].

Výukové programy byly tvořeny pro zadané aplikace "*sledování čáry*" (viz *Příloha č. 1*) a "*minisumo*" (viz *Příloha č. 2*) z důvodu použití jako demonstračních programu v manuálu o programování kontroléru PICAXE-28X1 ovládajícím mobilní robot UMU28A. Vzhledem k tomuto faktu byl při tvorbě programů bran zřetel na jednoduchost a přehlednost. Po napsání programu pro příslušnou aplikaci byl program testován v zadaném prostředí *Obr. 21*, *Obr. 28*. Pozorováním při testování programu byly zjišťovány a eliminovány příslušné chyby.

Pro úlohy jsou programy členěny na části dle toho, které se nejčastěji opakují, což znamená, že jsou vytvořeny podprogramy pro měření pomocí dvou druhů senzoru infračerveného záření a podprogram pro sériovou komunikaci s kontrolérem 16F866, což je kontrolér ovládající motory mobilního robotu. Programy pak využívají nadefinovaných podprogramu, čímž šetří místo a stávají se přehlednějšími, což je jedním z cílů. Každá aplikace pak obsahuje hlavní cyklicky se vykonávající část programu využívající těchto podprogramů pomocí "*volání*" (vnoření).

V úloze řešící sledování čáry bylo použito jen dvoustupňové reakce, jelikož se robot pohyboval nižší než poloviční rychlostí činící cca 0,17 *m/s*. Přesnější řízení rychlosti pohybu nebylo možné. Kontrolér PIC 16F688 nereagoval pozorovatelnou změnou rychlosti na malé odchylky řídicích hodnot od určitých zjištěných. Pomalejší pohyb mobilního robotu v této aplikaci byl volen také z důvodu časové prodlevy potřebné k tomu, aby proběhla komunikace mezi kontroléry PICAXE-28X1 a již zmíněným kontrolérem řídicím motory mobilního robotu, což opožďuje reakci robotu na informace o poloze senzorů nad čárou a tím vzniká možnost vyjetí zcela mimo čáru.

V úloze řešící zápas dvou robotů přetlačováním nebylo použito žádné zefektivnění schopnosti vytlačení soupeře z arény užitím *"manévrů"*. Po detekci okraje arény se robot snaží navrátit do arény a dojde-li k přejetí okraje arény robot, nedokáže rozpoznat, že prohrál, jelikož nemá dostačující paměť pro proměnné. Okraje arény musely být rozšířeny, jelikož robot nestíhal reagovat na informace poskytnuté senzory QRD1114.

Programy splňují podmínky uvedené v zadání a jsou v rámci předpokladů popsaných v této práci funkční. Úlohy jsou navrženy s ohledem na jednoduchost a přehlednost. Způsob řízení robotu však je hrubý s omezenou přesnosti. Toto by se dalo řešit zavedením matematických výpočtů pro regulaci do bloku rozhodování v programu, avšak nebylo tak učiněno z důvodu nároku na nepostačující paměť pro proměnné kontroléru PICAXE-28X1. Zlepšení regulovatelnosti je možné dosáhnout užitím příkazu *freq*, který dokáže zvýšit frekvenci vnitřního oscilátoru kontroléru z 4*MHz* na 8*MHz* nebo 16*Mhz*, čímž by se zrychlily procesy a také zvětšila energetická spotřeba kontroléru.

V závěru mohu konstatovat, že všechny body zadání byly realizovány.

Seznam použitých pramenů

[1] <URL: http://shop.snailinstruments.com/ >
[2] <URL: http://www.hobbyrobot.cz/ >
[3] <URL: http://www.fairchildsemi.com/>
[4] <URL: http://shop.snailinstruments.com/>
[5] <URL: http://www.rev-ed.co.uk/ >
[6] <URL: http://www.snailinstruments.com/ >
[7] <URL: http://shop.snailinstruments.com/ >
[8] <URL: http://cs.wikipedia.org/ >
[9] <URL: http://ii.fmph.uniba.sk/ >
[10] <URL: http://cs.wikipedia.org/ >
[11] <URL: http://cs.wikipedia.org/ >
[12] <URL: http://www.hobbyrobot.cz/ >

Seznam příloh

Příloha č. 1

```
#picaxe          28x1    ; Nastavení direktivy (Definice programovaného kontroléru).
; ------------------------------------------------------------------------------------------------------------ ;
DEFINICE PROMENNÝCH, V/V, SENZORŮ A PAMĚTI.
; ------------------------------------------------------------------------------------------------------------
symbol SL =    1        ; Přiražení symbolu VÝSTUPU (OUT1) pro spodní LED (zlepší
                                ; rozeznávaní barev).
symbol LB =    2        ; Přiražení symbolu VSTUPU (ADC2) pro spodní levý fototranzistor
                                ; (převedená hodnota).
symbol MB =    0        ; Přiražení symbolu VSTUPU (ADC0) pro spodní prostřední
                                ; fototranzistor (převedená hodnota).
symbol RB =     1       ; Přiražení symbolu VSTUPU (ADC1) pro spodní pravý fototranzistor
                                ; (převedená hodnota).
symbol LBP=    B0       ; Přiražení paměti pro odměřenou hodnotu z levého spodního
                                ; senzoru v čase t.
symbol MBP=    B1       ; Přiražení paměti pro odměřenou hodnotu z prostředního spodního
                                ; senzoru v čase t.
symbol RBP=     B2      ; Přiražení paměti pro odměřenou hodnotu z pravého spodního
                                ; senzoru v čase t.
symbol LBPp1= B3        ; Přiražení paměti pro odměřenou hodnotu z levého spodního
                                ; senzoru v čase t+1.
symbol MBPp1= B4        ; Přiražení paměti pro odměřenou hodnotu z prostředního spodního
                                ; senzoru v čase t+1.
symbol RBPp1= B5        ; Přiražení paměti pro odměřenou hodnotu z pravého spodního
                                ; senzoru v čase t+1.
; POZN.: PWM modulace o frekvenci 38kHz s proměnným plněním (vyžaduje SFH5110
        ; =>značná stabilita).
;-------------------------------------------------------------------------------------------------------------
; DEFINICE PROMENNÝCH A V/V PRO ŘIZENÍ MOTORŮ MOBILNÍHO ROBOTU
;-------------------------------------------------------------------------------------------------------------
symbol MT =    6        ; Definice VÝSTUPU sériové linky - přiražené pro motor (SERIN).
symbol RMV= B6          ; Definování PAMĚTI pro, určení a archivací hodnot vektoru rychlosti
                                ; (velikost, směr) pro pravý motor.
symbol LMV= B7          ; Definování PAMĚTI pro, určení a archivací hodnot vektoru rychlosti
                                ; (velikost, směr) pro levý motor.
```

```
; ------------------------------------------------------------------------------------------------------------
; PROGRAM PRO DETEKCI ČÁRY (ROZLIŠOVÁNÍ BAREV) A JEJÍ SLEDOVÁNÍ
; ------------------------------------------------------------------------------------------------------------
LBP    = 0              ; Počáteční čištění a nastavení paměti při startu programu.
MBP    = 0
RBP    = 0
LBPp1  = 0
MBPp1= 0
RBPp1  = 0
RMV    = 190            ; Hodnoty pro, zastavení motoru.
LMV    = 0
gosub MOTORC            ; Volání podprogramu řízení motorů.

RIDING:
                        ; Návěstí části programu pro řízení robotu.
gosub   BOTTOMSp1       ; Volání podprogramu pro detekci čáry BOTTOMSp1.
if       LBP <211 and MBP >236 and RBP <204        then
                                                  gosub STRAIGHT
                        ; Je-li robot prostředním senzorem na čáře, pak se volá podprogram
                                ; STRAIGHT.
elseif   LBP <207 and MBP >223 and RBP >236        then
                                                  RMV = 200 LMV = 93
                        ; Je-li robot levým senzorem mimo čáru a prostředním na okraji čáry
                                ; pak se nastaví hodnoty pro, řízení motorů takové aby robot
                                ; zatáčel (zlehka) doprava.
elseif   LBP <201 and MBP <166 and RBP >240        then
                                                  RMV = 197 LMV = 83
                        ; Je-li robot levým a prostředním senzorem mimo čáru pak se nastaví
                                ; hodnoty pro řízení motorů takové aby robot zatáčel
                                ; (středně) doprava.
elseif LBP >233 and MBP >207 and RBP <197 then
                                                  RMV = 209 LMV = 73
                        ; Je-li robot pravým senzorem mimo čáru a prostředním na okraji
                                ; čáry pak se nastaví hodnoty pro, řízeni motorů takové aby
                                ; robot zatáčel (zlehka) doprava.
elseif LBP >238 and MBP <173 and RBP <195 then
                                                  RMV = 220 LMV = 69
                        ; Je-li robot pravým a prostředním senzorem mimo čáru pak se
                                ; nastaví hodnoty pro řízení motorů takové, aby robot zatáčel
                                ; (středně) doprava.
endif                   ; Konec bloku rozhodování.
gosub MOTORC            ; Volání podprogramu MOTORC.
goto     RIDING         ; Skok na navěstí RIDING.
```

```
;-----------------------------------------------------------------------------------------------------------
; PODPROGRAMY
;-----------------------------------------------------------------------------------------------------------
BOTTOMSp1:
                        ; Podprogram detekce čáry (pro čas t a t+1).
        high            SL
                        ; Zapnuti spodních LED (zvýšeni citlivosti fototranzistoru).
        readadc LB, LBP
                        ; Čteni a zápis do paměti z levého spodního fototranzistoru.
        readadc MB, MBP
                        ; Čteni a zápis do paměti z prostř. spodního fototranzistoru.
        readadc RB, RBP
                        ; Čteni a zápis do paměti z pravého spodního fototranzistoru.
        pause           1
                        ; Časová prodleva mezi měřeními, za kterou robot urazí malou
                                ; vzdálenost, rozdíly mezi měřeními se vyhodnotí
                                ;(vyjádří polohu robotu vůči čáře).
        readadc LB,LBPp1
                        ; Čteni a zápis do paměti z levého spodního fototranzistoru (t+1).
        readadc MB,MBPp1
                        ; Čteni a zápis do paměti z prostř. spodního fototranzistoru (t+1).
        readadc RB,RBPp1
                        ; Čteni a zápis do paměti z pravého spodního fototranzistoru (t+1).
        low             SL
                        ; Vypnuti spodní LED za účelem šetřeni energie.
        return          ; Návrat na místo voláni podprogramu.

MOTORC:
                        ; Řízeni motoru, posláni hodnot proměnné xMV sériovou linkou
                                ;(příkazy pro motory).
        serout MT,T2400_4,(LMV)
        serout MT,T2400_4,(RMV)
        return          ; Návrat na místo voláni podprogramu.

STRAIGHT:
                        ; Podprogram vyhodnocující polohu robotu vůči čáře při jízdě
                                ;(minimalizace výchylek při jízdě).
         if       LBPp1 <207 or MBPp1 >223 or RBPp1 >236 then  RMV = 209 LMV = 93
                        ; Pokud předtím robot nezatáčel tak se nastaví hodnoty řízeni pro
                                ; jízdu rovně.
         elseif   LBPp1 >233 or MBPp1 >207 or RBPp1 <197 then  RMV = 220 LMV = 83
                        ; Pokud předtím robot zatáčel (doprava) tak se nastaví hodnoty řízeni
                                ; pro, zatáčeni na opačnou stranu (doleva).
         elseif   LBPp1 <211 or MBPp1 <236 or RBPp1 >204 then  RMV = 220 LMV = 93
                        ; Pokud předtím robot zatáčel (doleva) tak se nastaví hodnoty řízeni
                                ; pro, zatáčeni na opačnou stranu (doprava).
        endif           ; Konec bloku rozhodování.
return                  ; Návrat na místo voláni podprogramu.
;-----------------------------------------------------------------------------------------------------------
; KONEC PROGRAMU
;-----------------------------------------------------------------------------------------------------------
end                     ; Ukončení programu.
```

Příloha č. 2

```
#picaxe          28x1    ; Nastavení direktivy (Definice programovaného kontroléru).
; ----------------------------------------------------------------------------------------------------------
; DEFINICE PROMENNÝCH, V/V, SENZORŮ A PAMĚTI.
; ----------------------------------------------------------------------------------------------------------
symbol  IRS  = pin5     ; Definice VSTUPU IR-senzoru (IN5 -> SFH5110).
symbol  PWMV = 1        ; Definice VÝSTUPU pro PWM (PWM1 -> port C).
symbol  PWMP = 2        ; Definice VÝSTUPU pro, stanoveni pravé modulované
                                ;IR-LED (OUT2).
symbol LCNT  = B0       ; Přiřazení pamětí pro, čítaní odražených pulsů z levé strany (Left
                                ; Counter)
symbol RCNT  = B1       ; Přiřazení pamětí pro, čítaní odražených pulsů z pravé strany (Right
                                ; Counter)
symbol LCNTp1=B11; Přiřazení pamětí pro, čítaní odražených pulsů z levé strany (Left
                                ; Counter), v čase t+1.
symbol RCNTp1=B12; Přiřazení pamětí pro, čítaní odražených pulsů z pravé strany (Right
                                ; Counter), v čase t+1.
symbol MCNT= B2         ; Přiřazení pamětí pro, čítaní počtu měření senzorem SFH5110.
symbol SL = 1           ; Přiražení symbolu VÝSTUPU (OUT1) pro spodní LED foto-senzorů
                                ; (zlepší rozeznávaní barev).
symbol LB = 2           ; Přiražení symbolu VSTUPU (ADC2) pro spodní levý fototranzistor
                                ; (převedená hodnota).
symbol MB = 0           ; Přiražení symbolu VSTUPU (ADC0) pro spodní prostřední
                                ; fototranzistor (převedená hodnota).
symbol RB = 1           ; Přiražení symbolu VSTUPU (ADC1) pro spodní pravý fototranzistor
                                ; (převedená hodnota).
symbol LBP= B3          ; Přiražení paměti pro odměřenou hodnotu z levého spodního
                                ; senzoru v čase t.
symbol MBP= B4          ; Přiražení paměti pro odměřenou hodnotu z prostředního spodního
                                ; senzoru v čase t.
symbol RBP= B5          ; Přiražení paměti pro odměřenou hodnotu z pravého spodního
                                ; senzoru v čase t.
; POZN.: PWM modulace o frekvenci 38kHz s proměnným plněním (vyžaduje SFH5110
        ; =>značná stabilita).
;----------------------------------------------------------------------------------------------------------
; DEFINICE PROMENNÝCH A V/V PRO ŘIZENÍ MOTORŮ MOBILNÍHO ROBOTU
;----------------------------------------------------------------------------------------------------------
symbol MT = 6           ; Definice VÝSTUPU sériové linky - přiražené pro motor (SERIN).
symbol RMV= B6          ; Definování PAMĚTI pro, určení a archivací hodnot vektoru rychlosti
                                ; (velikost, směr) pro pravý motor.
symbol LMV= B7          ; Definování PAMĚTI pro, určení a archivací hodnot vektoru rychlosti
                                ; (velikost, směr) pro levý motor.
```

```
; ------------------------------------------------------------------------------------------------------------
; PROGRAM PRO ZÁPAS V APLIKACI MINISUMO
; ------------------------------------------------------------------------------------------------------------
LBP    = 0              ; Počáteční čištění a nastavení paměti při startu programu.
MBP    = 0
RBP    = 0
LBPp1  = 0
MBPp1= 0
RBPp1  = 0
RMV    = 190            ; Hodnoty pro, zastavení motorů.
LMV    = 0
gosub  MOTORC           ; Volání podprogramu řízení motorů.
wait   4                ; Časová prodleva 4 s (čekáni na začátek zápasu).
RMV    = 215            ; Hodnoty pro, pohyb vpřed (začátek zápasu).
LMV    = 85
gosub  MOTORC           ; Volání podprogramu řízení motorů.

FIGHT:
                        ; Návěstí části programu pro řízení robotu.
gosub  FRONTS           ; Volání podprogramu měření (předním) senzorem SFH5110.
if     LCNTp1>3 or  RCNTp1>3      then
                                   gosub   FOLLOW
                        ; Jestli je detekován soupeř, tak se volá podprogram pro
                                ; pronásledování FOLLOW.
elseif LCNTp1< 1 and RCNTp1< 1   then
                                   LMV= 75        RMV= 200
                        ; Jestli není detekována překážka (soupeř), tak se nastaví řídicí
                                ; hodnoty pro pomalou jízdu vpřed.
endif                   ; Konec bloku rozhodování.
gosub  BOTTOMS          ; Volání podprogramu měření (spodními) senzory QRD1114.
gosub  BORDERS          ; Volání podprogramu kontroly najetí na okraj arény.
gosub  MOTORC           ; Volání podprogramu řízení motorů.
goto   FIGHT            ; Skok na navěstí FIGHT (Zápas).
```

```
;-----------------------------------------------------------------------------------------------------------
; PODPROGRAMY
;-----------------------------------------------------------------------------------------------------------
        FRONTS:
                        ; Návěští podprogramu měření senzorem SFH5110 (detekce překážky).
        LCNT  =0        ; Počáteční čištění a nastavení paměti při startu měření.
        MCNT =0
        RCNT  =0
        LCNTp1=0
        RCNTp1=0
                        ; MĚŘENÍ V ČASE t.
        for MCNT = 1 to 4
                        ; Zajištěni 4 opakování měřeni (volitelné – ovlivňuje celkovou dobu
                                ; měřeni a tím i zpoždění reakce).
                LPIRL:
                        ; Nevěští měřeni odezvy (odrazu) z levé IR-LED.
                low             PWMP
                        ; Nastavení log 0 na výstup OUT2 pro, umožněni PWM regulace
                                ; levé IR LED (viz schéma zapojení).
                pwmout          PWMV,26,26
                        ; Zapnuti PWM při f=38 kHz, plnění 25%.
                        ; (Duty Cycle - doba periody po kterou signál bude v log.1).
; POZN.: Výpočet hodnot pro plnění PWM: viz PICAXE – WIZARDS - pwmout...
                pauseus         100
                        ; Doba měřeni IR-senzoru (SFH5110) je 1ms.
                if      IRS=1   then
                                LCNT=LCNT
                        ; Pokud IR-senzor posila do procesoru log 1, pak se do paměti uloží
                                ; (hodnota +0) že není detekována žádna překážka,
                                else
                                LCNT=LCNT+1
                                ; v opačném případě (IR-senzor posila log 0) se do paměti uloží,
                                ; že je detekována překážka (hodnota +1).
                endif   ; Konec podmíněného větveni programu.
                pwmout          PWMV,255,0
                        ; Zastavení (vypnuti) PWM (Duty Cycle -> 0%).
                pause           9
                        ; Časová prodleva (9ms) - obnoveni citlivosti IR-senzoru (SFH5110).

                PPIRL:
                        ; Nevěští měřeni odezvy (odrazu) z pravé IR-LED.
                high            PWMP
                        ; Nastavení log 1 na výstup OUT2 pro, umožněni PWM regulace
                                ; pravé IR LED (viz schéma zapojení).
                pwmout          PWMV,26,78
                        ; Zapnuti PWM při f=38 kHz, plnění 75%.
                                ; (Duty Cycle - doba periody, po kterou signál bude v log.1).
                pauseus         100
                        ; Doba měřeni IR-senzoru (SFH5110) je 1ms.
                if      IRS=1   then
                                RCNT=RCNT
                        ; Pokud IR-senzor posila do procesoru log 1, pak se do paměti uloží
                                ; (hodnota +0) že není detekována žádna překážka,
```

```
                    else
                    RCNT=RCNT+1
                    ; v opačném případě (IR-senzor posila log 0) se do paměti uloží,
                    ; že je detekována překážka (hodnota +1).
            endif   ; Konec podmíněného větveni programu.
            pwmout      PWMV,255,0
                ; Zastaveni (vypnuti) PWM (Duty Cycle -> 0%).
            pause       9
                ; Časová prodleva (9ms) - obnoveni citlivosti IR-senzoru (SFH5110).
            low         PWMP
                ; Nastavení log 0 na výstup OUT2 pro, šetřeni energie.
        next    MCNT ; Inkrementace (zvýšeni hodnoty +1) hodnoty počtu opakování měřeni.
        pause       50
                ; Časová prodleva mezi měřeními.
        MCNT  = 0   ; Vynulování hodnoty počtu opakování měřeni (příprava na další
                    ; smyčku měření).

                ; MĚŘENÍ V ČASE t+1.
        for MCNT = 1 to 4
                ; Zajištěni 4 opakování měřeni (volitelné – ovlivňuje celkovou dobu
                    ; měřeni a tím i zpoždění reakce).

            LPIRLp1:
                ; Nevěští měřeni odezvy (odrazu) z levé IR-LED.
            low         PWMP
                ; Nastavení log 0 na výstup OUT2 pro, umožněni PWM regulace
                    ; levé IR LED (viz schéma zapojení).
            pwmout      PWMV,26,26
                ; Zapnuti PWM při f=38 kHz, plnění 25%.
                ; (Duty Cycle - doba periody, po kterou signál bude v log.1).
; POZN.: Výpočet hodnot pro plnění PWM: viz PICAXE – WIZARDS - pwmout...
pauseus     100
                ; Doba měřeni IR-senzoru (SFH5110) je 1ms.
            if      IRS=1  then
                    LCNTp1=LCNTp1
                ; Pokud IR-senzor posila do procesoru log 1, pak se do paměti uloží
                    ; (hodnota +0) že není detekována žádna překážka,
                    else
                    LCNTp1=LCNTp1+1
                    ; v opačném případě (IR-senzor posila log 0) se do paměti uloží,
                    ; že je detekována překážka (hodnota +1).
            endif   ; Konec podmíněného větveni programu.
            pwmout      PWMV,255,0
                ; Zastaveni (vypnuti) PWM (Duty Cycle -> 0%).
            pause       9
                ; Časová prodleva (9ms) - obnoveni citlivosti IR-senzoru (SFH5110).
            PPIRLp1:
                ; Nevěští měřeni odezvy (odrazu) z pravé IR-LED.
            high        PWMP
                ; Nastavení log 1 na výstup OUT2 pro, umožněni PWM regulace
                    ; pravé IR LED (viz schéma zapojení).
            pwmout      PWMV,26,78
                ; Zapnuti PWM při f=38 kHz, plnění 75%.
```

```
                    ; (Duty Cycle - doba periody po kterou signál bude v log.1).
          pauseus   100
                ; Doba měřeni IR-senzoru (SFH5110) je 1ms.
          if    IRS=1  then
                    RCNTp1=RCNTp1
                ; Pokud IR-senzor posila do procesoru log 1, pak se do paměti uloží
                    ; (hodnota +0) že není detekována žádna překážka,
                    else
                    RCNTp1=RCNTp1+1
                    ; v opačném případě (IR-senzor posila log 0) se do paměti uloží,
                    ; že je detekována překážka (hodnota +1).
          endif ; Konec podmíněného větveni programu.
          pwmout    PWMV,255,0
                ; Zastaveni (vypnuti) PWM (Duty Cycle -> 0%).
          pause     9
                ; Časová prodleva (9ms) - obnoveni citlivosti IR-senzoru (SFH5110).
          low       PWMP
                ; Nastavení log 0 na výstup OUT2 pro, šetřeni energie.
    next  MCNT ; Inkrementace (zvýšeni hodnoty +1) hodnoty počtu opakování měřeni.
    return      ; Návrat na místo voláni podprogramu.


FOLLOW:
                ; Nevěští části programu pro, následování soupeře.
    if      LCNT<1 and  RCNT<1      then
                                    LMV= 85     RMV= 215
                ; Jestli je soupeř přímo před robotkem (střední vzdálenost – víc než
                    ; 5cm) tak se nastaví řídicí hodnoty pro středně rychlý pohyb
                    ; vpřed.
    elseif  LCNT>1 and  RCNT>1      then
                                    LMV= 120    RMV= 245
                ; Jestli je soupeř přímo před robotkem (malá vzdálenost – míň než 5cm)
                    ; tak se nastaví řídicí hodnoty pro rychlý pohyb vpřed.
    elseif  LCNT<1 and  RCNT>1      then
                                    LMV= 125    RMV= 210
                ; Jestli je soupeř vpředu zprava před robotkem tak se nastaví řídicí
                    ; hodnoty pro zatáčení doprava.
    elseif LCNT>1 and   RCNT<1      then
                                    LMV= 83     RMV= 250
                ; Jestli je soupeř vpředu zleva před robotkem tak se nastaví řídicí
                    ; hodnoty pro zatáčení doleva.
    endif       ; Konec podmíněného větveni programu.
    return      ; Návrat na místo volání podprogramu.
```

```
BORDERS:
                    ; Nevěští části programu pro, kontrolu najetí na okraj arény.
        if  LBP<60 and MBP<60 and RBP<60        then
                                                LMV= 0      RMV= 190
                    ; Jestli robot najel na bílý okraj arény tak se nastaví řídicí hodnoty
                        ; pro zastavení.
                                                gosub MOTORC
                    ; Volání podprogramu řízení motorů.
                                                LMV= 32     RMV= 161
                    ; Nastavení řídicích hodnot pro pohyb dozadu (přímočarý).
                                                gosub MOTORC
                    ; Volání podprogramu řízení motorů.
                                                pause 600
                    ; Časová prodleva 600 ms, aby robot couvnul dostatečně do arény.
                                                LMV= 125    RMV= 155
                    ; Nastavení řídicích hodnot pro rotaci robotu doprava.
        endif       ; Konec podmíněného větveni programu.
        return      ; Návrat na místo voláni podprogramu.


BOTTOMS:
                    ; Podprogram detekce čáry.
        high        SL
                    ; Zapnutí spodních LED (zvýšeni citlivosti fototranzistoru).
        readadc LB, LBP
                    ; Čtení a zápis do paměti z levého spodního fototranzistoru.
        readadc MB, MBP
                    ; Čtení a zápis do paměti z prostř. spodního fototranzistoru.
        readadc RB, RBP
                    ; Čtení a zápis do paměti z pravého spodního fototranzistoru.
        low         SL
                    ; Vypnuti spodní LED za účelem šetřeni energie.
        return      ; Návrat na místo volání podprogramu.


MOTORC:
                    ; Řízení motorů, poslání hodnot proměnné xMV sériovou linkou
                        ;(příkazy pro motory).
        serout MT,T2400_4,(LMV)
        serout MT,T2400_4,(RMV)
        return      ; Návrat na místo volání podprogramu.

;-----------------------------------------------------------------------------------------------------------
; KONEC PROGRAMU
;-----------------------------------------------------------------------------------------------------------
end                 ; Ukončení programu.
```